

Yüksek Lisans Tezi
Trakya Üniversitesi Fen Bilimleri Enstitüsü

E-POSTA METİNLERİ ÜZERİNDE İÇERİK TARAMASI YAPAN BİR UYGULAMA GELİŞTİRİLMESİ

ÖZET

Bu çalışmanın amacı, uzak posta sunucusuna bağlanıp POP3 protokolünü kullanarak, bir kullanıcının hesabına gelen postalar üzerinde, belirlediği bir sorgu ifadesini uygulayıp, sorguya uyanlar ve uymayanlar olarak ayıran ve bunu kullanıcıya bildiren istemci bir uygulamanın geliştirilmesidir.

Çalışma, temel olarak iki ana parçadan oluşmaktadır. Birincisi, herhangi bir POP3 sunucusuyla iletişim kuran bir istemci ağ uygulaması olmasıdır. Bu amaçla, Internet protokollerinin ve standartlarının yapısı, çalışma biçimleri, geliştirilme süreçleri incelenmiş, POP3 protokolü üzerinde detaylı bir şekilde durulmuştur. İkincisi ise, anahtar kelime tabanlı mantıksal sorgulara dayalı bir arama mekanizması kullanıldığından çalışma, küçük bir “Bilgi Erişim” uygulamasına giriş olarak da değerlendirilebilir. Bu amaçla, bilgisayar bilimlerinin bu alanı üzerinde de çalışma yapıp uygulaması gerçekleştirilmiştir.

Anahtar Kelimeler: POP3, Internet Protokolleri, Bilgi Erişim Sistemleri, Veri Erişim Sistemleri

Yıl: 2005

Sayfa : 58

Master Thesis
Trakya University Graduate School of
Natural and Applied Sciences
Department of Computer Engineering

DEVELOPING AN APPLICATION THAT MAKES CONTENT-SCANNING ON E-MAIL TEXTS

SUMMARY

The aim of this study is to develop a client application that connects a remote mail server using the POP3 protocol, executes a query term which described by the user in the incoming mails of mailbox, classifies these mails as “match to query” or “not match to query” and notifies the user about these query results.

The study is comprises two parts, the part is client a network application which enables communication with any POP3 server. For this purpose, the Internet protocols and their standards structures, their working conditions and improvement processes have been investigated the POP3 protocol has been analyzed in detail.

The second part of the study is evaluated as an introduction to the “Information Retrieval” because of using the searching mechanism which is supported by keyword-based logical queries. Therefore, this field of computer sciences has also been studied and its application has been realized.

Key Words: POP3, Protocols, Information Retrieval, Data Retrieval

Year:2005

Page:58

TEŞEKKÜR

Bu çalışmanın gerçekleşmesinde, değerli katkılarından dolayı tez danışmanım Sayın Yrd. Doç. Dr. Erdem UÇAR'a, Trakya Üniversitesi Bilgisayar Mühendisliği Bölümü Öğretim Üyesi Sayın Yrd.Doç.Dr.Yılmaz KILIÇASLAN'a teşekkür ederim.

Ayrıca çalışma sürecince desteklerini esirgemeyen Arş.Gör.Emrah OĞUZHAN' a, Öznur GAZAZ'a, Arş.Gör.Levent VURAL'a, Arş.Gör.Serbülent PAKSUZ'a ve Arş.Gör.Şahin DÜNDAR'a da teşekkür ederim.

Son olarak da, çalışma süresince her zaman yanımda olan aileme de teşekkür ederim.

1. GİRİŞ VE AMAÇ	1
2. ve ELEKTRONİK POSTA	2
2.1 Internet'in ve Elektronik Postanın Doğuşu	2
2.2 Internet Standartlarının Özellikleri ve Geliştirilmesi	4
2.3 Internet Protokollerinin Genel Yapısı	7
2.3.1 Bağlantısız Protokoller ve Bağlantı Protokoller	7
2.3.2 Durumsuz Protokoller ve Durumlu Protokoller	8
2.4 Elektronik Posta Erişim Paradigmaları	9
2.5 Elektronik Posta Dökümanlarının Yapısı	11
3. POP3 PROTOKOLÜ	15
3.1 Tarihçe	15
3.2 Genel Bakış	16
3.3 Durumlar	17
3.3.1 Yetkilendirme Durumu	18
3.3.2 İşlem Durumu	19
3.3.3 Güncelleme Durumu	19
3.4 Komutlar	20
3.4.1 USER Komutu	20
3.4.2 PASS Komutu	21
3.4.3 APOP Komutu	22
3.4.4 STAT Komutu	23
3.4.5 LIST Komutu	23
3.4.6 RETR Komutu	24
3.4.7 DELE Komutu	26
3.4.8 RSET Komutu	27
3.4.9 QUIT Komutu	27
3.4.10 Diğer Komutlar	27
4. SMTP PROTOKOLÜ	28
4.1.Genel Bakış	28
4.2 Temel Komutlar	29
4.2.1 HELO Komutu	29
4.2.2 MAIL Komutu	29
4.2.3 RCPT Komutu	30
4.2.4 DATA Komutu	31
4.2.5 RSET Komutu	31
4.2.6 QUIT Komutu	32
5. KLASİK BİLGİ ERİŞİM SİSTEMİ MODELLERİ	33
5.1 Temel Kavramlar	33
5.2 Mantıksal Model	34
5.3 Vektör Model	35
5.4 Olasılık Yaklaşımlı Model	36
6. YAZILIMIN GERÇEKLEŞTİRİLMESİ	38
6.1 Genel Tanıtım	38
6.2 Yazılımın Genel Veri Akış Diyagramı	39
6.3 POP3 Sunucusuna Bağlanmak Ve Elektronik Postaların Alınması	40

6.4 Elektronik Posta Metinleri Üzerinde Sorgunun İşletilmesi	45
6.5 Sonuçların Kullanıcıya Gösterilmesi.....	47
7. TARTIŞMA VE SONUÇ.....	48
KAYNAKLAR	49

ŞEKİLLER LİSTESİ

Şekil 2.1 Offline Model’de Elektronik Posta Gönderme-Alma Süreci.....	10
Şekil 2.2 Bir Elektronik Posta metninin iç yapısı.....	12
Şekil 2.3 Eklentili Elektronik Posta Metni.....	13
Şekil 2.4 Çok Parçalı Bir Elektronik Posta Metninin İç Formatı.....	14
Şekil 3.1 İstemci ile Sunucu arasındaki POP3 Oturumundaki Durumlar.....	18
Şekil 3.2 İstemcinin USER komutuna sunucunun verdiği yanıt.....	20
Şekil 3.3 PASS Komutuna sunucunun gönderdiği olumlu ve olumsuz yanıtlar.....	21
Şekil 3.4 STAT Komutu ve Sunucu tarafından gönderilen yanıt.....	23
Şekil 3.5 LIST Komutu ve Sunucu tarafından gönderilen yanıt.....	24
Şekil 3.6 RETR Komutuna sunucudan yanıt ve elektronik postanın içeriği.....	25
Şekil 4.1 HELO komutu ve sunucu tarafından gönderilen yanıt.....	29
Şekil 4.2 MAIL komutu ve sunucu tarafından gönderilen yanıt.....	30
Şekil 4.3 RCPT komutu ve gönderilen olumlu yanıt.....	30
Şekil 4.4 DATA komutu ve sunucudan gelen yanıtlar.....	31
Şekil 6.1 Yazılımın Veri Akış Diyagramı.....	39
Şekil 6.2 Sunucuyla POP3 protokolü ile iletişim kuran Visual Basic ile yazılmış fonksiyon.....	41
Şekil 6.3 Sunucu yanıtını kontrol eden Visual Basic ile yazılmış fonksiyon.....	43
Şekil 6.4 Yazılım içerisinde sorgu ifadesinin bir elektronik posta metnine nasıl uygulandığını gösteren Visual Basic kaynak kodu.....	46

TABLÖLAR LİSTESİ

Tablo 2.1 Elektronik Posta Erişim Modellerinin karşılaştırılması.....	11
Tablo 6.1 Mantıksal İşleçler ve Yazılımdaki karşılıkları.....	45

1. GİRİŞ VE AMAÇ

Bilindiği gibi Internet'in ortaya çıkması ve son yıllarda hızla yaygınlaşmasıyla, kullanıcıların bilgiye olan erişimi daha önce hiç olmadığı kadar kolaylaşmıştır. Fakat Internet ortamı aynı zamanda iyi yapılandırılmamış ve düzensiz bir doküman yığına sahiptir. Kullanıcıların bu doküman yığını içerisinde, isteklerine uygun olanları, en kısa sürede bulup çıkarmalarını sağlama çalışmaları özellikle son yıllarda son derece önem kazanmıştır. Internet arama motorları bu amaçla geliştirilmiş uygulamalardır ve son birkaç yıldır Internet kullanıcılarının bilgiye erişimde kullandıkları vazgeçilmez araçlardan olmuştur.

Internet'in yaygınlaşması diğer taraftan insanların birbirleriyle olan yazılı iletişimini de etkilemiştir. Günümüzde, normal posta yerine, Internet ortamında yol alan elektronik postalar (e-posta), bildiğimiz posta kutuları yerine Internet ortamında oluşturulmuş sanal posta kutuları kullanılmaktadır. Elektronik postanın, dünyanın bir ucundan diğer ucuna saniyelerle ölçülen sürelerde ulaşması beraberinde başka sorunları da getirmiştir. Kullanıcılar, kutularında her gün onlarca hatta yüzlerce elektronik posta görür duruma gelmişlerdir.

Bu çalışmada amacımız, kullanıcının, sunucudaki posta kutusunda saklanan elektronik postaları üzerinde, belirlediği sorgu ifadesine göre sınıflandırma yapmasını sağlayan bir yazılımın gerçekleştirilmesidir. Böylece, ilgi sahamıza, amacımıza ve çalışma alanımıza uyan veya uymayan e-postaları ayırmak mümkün olacaktır. Çalışmanın ilk bölümünde Internet ve e-postanın özellikleri üzerinde durulmuş, Internet standartları ve protokollerinin geliştirilme sürecinden bahsedilmiştir. İkinci bölümde e-posta alım protokolü üzerinde detaylı şekilde durulmuştur. Üçüncü bölümde ise gönderim protokolüne değinilmiştir. Dördüncü bölümde Klasik Bilgi Erişim Modelleri anlatılmıştır. Son olarak da, geliştirilen yazılımın hakkında bilgi verilmiştir.

2. INTERNET ve ELEKTRONİK POSTA

2.1 Internet'in ve Elektronik Postanın Doğuşu

Yirminci yüzyılın ortalarından itibaren bilgisayarların araştırma merkezleri ve üniversitelerde yaygınlaşması ile beraber bu cihazların birbirlerine bağlanarak iletişim kurması fikri ortaya çıkmıştır. MIT (Massachusetts Institute of Technology) araştırmacılarından J.C.R. Licklider 1962 yılında yayımladığı makalelerinde “Galaktik Ağ” kavramından söz eder. Bu makalelerde, insanların istediği herhangi bir yerden ve herhangi bir zamanda bilgiye kolayca ulaşabilmesi sağlayabilen bir yapının varlığı tartışılır. Licklider, 1962 yılında DARPA’ da (Defense Advanced Research Projects Agency) bir araştırma projesinin başına geçer. İlerleyen zamanlarda bilgisayarların birbirlerine bağlanması fikrinin önemini, çalışma arkadaşları Ivan Sutherland, Bob Taylor ve MIT araştırmacılarından Lawrence G. Roberts te farkederek. 1965'te Roberts, Thomas Merrill ile birlikte, MIT'teki bir TX2 bilgisayarı ile California'daki bir Q32 bilgisayarını düşük hızlı telefon hattı üzerinden birbirileri ile haberleştirmeyi başarırlar. Bu kurulan bağlantı ilk genişalan bilgisayar ağı olarak tarihe geçer. Bu deneyle birlikte, zaman paylaşımlı bilgisayarların birbirileri ile kolay bir şekilde verilerini paylaşabildikleri gösterilmiş olur.

1966 yılında Roberts, DARPA'da bilgisayar ağları kavramını geliştirmiştir. 1967 yılında bunu ARPANET (Advanced Research Projects Agency Network) adıyla yayınlar. Ağustos 1968'te Roberts ve DARPA üyeleri, ARPANET konusundaki spesifikasyonları belirledikten sonra DARPA tarafından Interface Message Processors (IMP) adıyla ilk paket anahtarlarını tanıtan bir RFQ (Request For Quotation) yayınlanılır. RFQ, Aralık 1968'te Frank Heart liderliğinde BBN' de (Bolt, Beranek ve Newman) kabul görür. BBN ekibi IMP'ler üzerinde çalışırken Bob Kahn, ARPANET'in mimari tasarımında önemli bir rol oynamıştır. Ağ topolojisi ve ekonomisi, Roberts ile birlikte çalışan Howard Frank ve Network Analysis Corporation ekibi tarafından tasarlanmış ve optimize edilmiştir.

1972'de ilk uygulama olan "elektronik posta" hizmeti de tanıtıldı. Mart ayında BBN'den Ray Tomlinson, elektronik posta için mesaj gönderme ve alma programlarını yazdı. Daha sonra Roberts, buna ek olarak bazı yeni uygulamalar (list, file, forward, respond gibi) geliştirmiştir. Bu olay, bugün gördüğümüz pek çok farklı uygulamanın da habercisi olmuştur (WWW gibi).

Elektronik mesaj iletimi başlangıçta FTP' nin (File Transfer Protocol) bir parçası olarak değerlendiriliyordu. Daha sonraları bu düşünce değişmeye başladı ve elektronik posta gönderimi ve alımı için bağımsız protokoller geliştirildi.

1981 yılına gelindiğinde posta gönderimi için SMTP (Simple Mail Transfer Protocol) protokolü J.Postel tarafından geliştirildi ve bu protokol RFC 788 ile tanımlandı.

1982'de Internet üzerinde yol alan elektronik mesajlar için ortak bir standart oluşturuldu. Bu format D.Crocker tarafından RFC 822'de tanımlandı.

1984 yılında J.K. Reynolds tarafından POP protokolü geliştirildi ve RFC 918 ile tanımlandı.

1992 ye gelinceye kadar elektronik postalar basit içerikli metin dizileri (ASCII dizileri) şeklindeydiler. N. Borenstein ve N. Freed, RFC 1341 ile MIME'yi tanımladı. MIME sayesinde artık elektronik postaya başka dosyalar (görüntü dosyaları, ses dosyaları, vb..) eklenip gönderilebiliyordu. Böylece elektronik postalarda içerik daha da zenginleşmiş oldu ve e-postalar daha esnek hale geldiler.

Internet'in son 10-15 yıldaki çok hızlı yaygınlaşmasına paralel olarak elektronik posta kullanımı da inanılmaz oranlarda artmıştır. Elektronik postanın çok sayıda kişi tarafından kullanılması ve ulaşım hızı, bazı sorunları da beraberinde getirdi. Örneğin şirketler ürünlerini veya hizmetlerini tanıtmada hızlı olduğu için ve bir anda milyonlarca kişiye birden ulaşabildiği için elektronik postayı kullanmaya başladılar. Kişiler her gün posta kutularında onlarca reklam e-postası görür oldu. Bu durum yeni bir kavramı ortaya

çıkardı; SPAM “İstenmeyen E-Posta”. Bazı hesaplamalara göre şu anda dünya üzerinde yollanan her üç e-postadan iki tanesi SPAM yani istenmeyen e-postadır ve bu oranın ileriki yıllarda daha da artacağı öngörülmektedir.

Günümüzde elektronik posta alanındaki araştırmalar devam etmektedir. Araştırmalar daha çok güvenlik ve “istenmeyen elektronik posta” sorununun çözümüne odaklanmış durumdadır.

2.2 Internet Standartlarının Özellikleri ve Geliştirilmesi

Internet, yapısı gereği bazı ortak standartların ve spesifikasyonların belirlenmesini zorunlu kılar. Bu Internet standartları ve özellikleri; standart kuruluşları ve bunlara bağlı çalışma grupları tarafından oluşturulurlar. Bu kuruluşlar yeni standartlar oluşturur veya var olan standartları ve özellikleri geliştirirler. Üç temel standart grubu Internet ile ilgili standartlar ve özellikler geliştirmektedir. Bu gruplar

- ISO – International Organization for Standardization
- IETF – Internet Engineering Task Force
- W3C – The World Wide Web Consortium

Uluslararası Standartlar Enstitüsü (ISO)

Uluslararası Standartlar Enstitüsü dünyadaki en büyük ve en önemli standart oluşturma kuruluşudur. ISO özel olarak Internet’le ilgili standartların geliştirilmesi ile ilgilenmez; daha geniş çerçevede, bilgisayar ağlarıyla ilgili standartlarla ilgilenir. ISO tarafından oluşturulmuş en bilinen ve uluslararası alanda kabul gören yedi katmanlı ağ modeline Open System Interconnection (OSI) Reference Model (Açık Sistemlerarası Bağlantılar Referans Modeli) adı verilir.

Bununla beraber ISO tarafından oluşturulmuş olan karakter setleri standartları yine aynı OSI modeli gibi uluslararası düzeyde kabul görmüştür ve tüm uygulamalarda kullanılmaktadır. Örnek olarak ISO 8859-1, ISO-Latin-1 karakter kümesini tanımlamak için kullanılır Türkçe için belirlenmiş karakter kümesi ISO-8859-9 dur.

Internet Mühendisliği Görev Timi

Internet Mühendisliği Görev Timi (IETF), Internet standartlarını geliştiren asıl kuruluştur. Mevcut Internet standartlarına yapılan değişiklikler ve teklifler IETF tarafından tasdiklenir. IETF, Internet için tarifnameleri belirlemek üzere yılda üç kez toplanır.

Mevcut özellikler için değişiklikler ve teklifler, değişiklikleri tartışmak ve sunmak için toplanan resmi komite tarafından tasdiklenir. Bu resmi komitelere çalışma grupları adı verilir. IETF’de düzinelerce çalışma grubu vardır. Genellikle her grup belli bir geliştirme alanında özel bir başlık üzerinde yoğunlaşır. Bazı geliştirme alanları şunlardır:

- Uygulamalar
- Internet Protokolü : Yeni Nesil
- Internet
- Ağ Yönetimi
- İşletme Gereksinimleri
- Yönlendirme
- Güvenlik
- Ulaştırma
- Kullanıcı Hizmetleri

Çalışma gruplarında, özellikleri değiştirmeyi tasdikleme süreci standartlaştırılmıştır.

Çalışma grupları Internet Draft (Taslak) özelliklerini sunarlar. Internet Taslağı resmileştikten altı ay sonra geçerlidir. Eğer Internet Taslağı altı ay içerisinde onaylanmazsa taslak zaman aşımına uğrar ve geçerliliğini yitirir. Eğer Internet Taslağı tasdiklenirse bir RFC (Request For Comment) olur.

RFC'ler sürekli arşivlenirler ve yerlerine yeni bir RFC gelene kadar geçerliliklerini devam ettirirler. RFC'ler isimlerinden de anlaşılabilceği gibi daha iyi hale getirilebilmek amacıyla tartışılması ve yeni öneriler getirilmesi için tüm Internet camiasına sunulmuştur.

RFC'lerin çoğu sonunda Internet standartları haline gelmektedir ancak süreç o kadar da hızlı değildir. Örneğin URL'ler ilk olarak 1990 da World Wide Web'e sunuldu. O günden beri kullanılmalarına rağmen 1994 Aralık ayına kadar RFC olamadılar ve ancak kısa bir süre önce Internet standardı olarak onaylandılar.

IETF'ye üyelik herkese açıktır. Çalışma gruplarının yöneticileri IETF'nin iç yönetimini üstlenirler. Bu yöneticiler IETF'nin Başkanı da dahil olmak üzere hep birlikte Internet Engineering Steering Group-IESG'i (Internet Mühendislik İdari Grubu) oluştururlar. Internet Society'nin (Internet Camiası) yönetimi altındaki IESG, IETF'nin işletme yönetimini üstlenmektedir.

World Wide Web Konsorsiyumu

World Wide Web konsorsiyumu Amerika Birleşik Devletlerinde bulunan MIT'de (Massachusetts Institute of Technology) Bilgisayar Bilimleri Laboratuvarı tarafından yönetilmektedir. W3 Konsorsiyumu World Wide Web' in gelişimi için ortak standartlar oluşturulmak amacıyla çalışmaktadır. MIT, CERN (European Organization for Nuclear Research) ve INRIA'nın (Institut National de Recherche en Informatique et en Automatique) birlikte başlattıkları bir konsorsiyumdur. ABD W3C, merkezi MIT tarafından kurulmuş ve çalıştırılmaktadır. Avrupa W3C merkezi INRIA'dadır. CERN ve INRIA, Avrupa W3C merkezinin yönetimi için işbirliği yapmaktadır.

W3C'nin kurulma amaçlarından biri de Web teknolojilerindeki gelişmeler için ortak standartlar oluşturulmasına yardımcı olmaktır. W3C'nin temel amaçlarından bir diğeri de kullanıcılara ve geliştiricilere Web ile ilgili bilgilerin bulunduğu bir depo sunmaktır. Bu amaç doğrultusunda W3C'nin Web gelişimi ile ilgili en yeni bilgileri içeren merkezleri bulunmaktadır.

W3C, IETF'ye özellik ve yeni standart teklifleri hazırlamak için üye kuruluşları ile birlikte çalışır. Üye kuruluşlar üyelik statülerine bağlı olarak bir miktar üyelik ücreti öderler.

2.3 İnternet Protokollerinin Genel Yapısı

2.3.1 Bağlantısız Protokoller ve Bağlantı Temelli Protokoller

Bağlantısız protokoller, bağlantı temelli protokollerden isteklere verilen cevaplar ve karşılıklar açısından ayrılırlar. Bağlantısız bir protokolde istemci sunucuya bağlanır, bir istekte bulunur, cevap alır ve ardından diğer isteklere hizmet vermek için bağlantı kurar.

Bağlantı temelli bir protokol örneği FTP'dir. Bir FTP sunucusuna bağlanıldığında dosyayı aktardıktan sonra bağlantı devam eder. Bu bağlantının tutulması için sistem kaynakları gereklidir. Çok sayıda açık bağlantısı bulunan bir sunucu kolaylıkla çökebilir. Sonuç olarak, FTP sunucularının çoğu belirli bir anda en fazla 250 bağlantıya izin verebilecek şekilde konfigüre edilmektedir. Bunun anlamı bir seferde ancak 250 kullanıcının sunucuya erişebileceğidir. Doğru şekilde bağlantısı sonlandırılmayan işlemler, sunucu üzerinde sorunlara neden olabilir. En kötüsü işlemler kontrolden çıkabilir ve hatta sunucu çökebilir. Bu işlemlerin kötü tarafı sistem kaynaklarını gasbetmeleridir.

FTP'nin tersine HTTP bağlantısız bir protokoldür. İstemciler sunucuya bağlandıklarında istekte bulunur, cevabını alır ve ardından bağlantı kesilir. Bağlantı saklı

tutulmadığı için işlem tamamlandığında hiçbir sistem kaynağı meşgul edilmez. Sonuç olarak HTTP sunucuları, aktif bağlantılarla sınırlıdır ve en az bir sistem yükü ile binlerce işleme hizmet verebilirler. Bağlantısız protokollerin dezavantajı ise aynı istemci bir kez daha istekte bulunduğu anda bağlantının yeniden kurulması gereklidir. Bu, kullanıcılar için her yeni bilgi istediğinde bir gecikme anlamına gelir (Stanec,1997).

2.3.2 Durumsuz Protokoller ve Durumlu Protokoller

HTTP durumu olmayan protokollere bir örnektir. Durumu olmayan protokoller durumlu protokollerden isteklerle ilgili bilgilerin tutuluş şeklinde farklılık gösterirler. Durumu olmayan protokollerde işlem işlendikten sonra durum bilgisi saklanmaz.

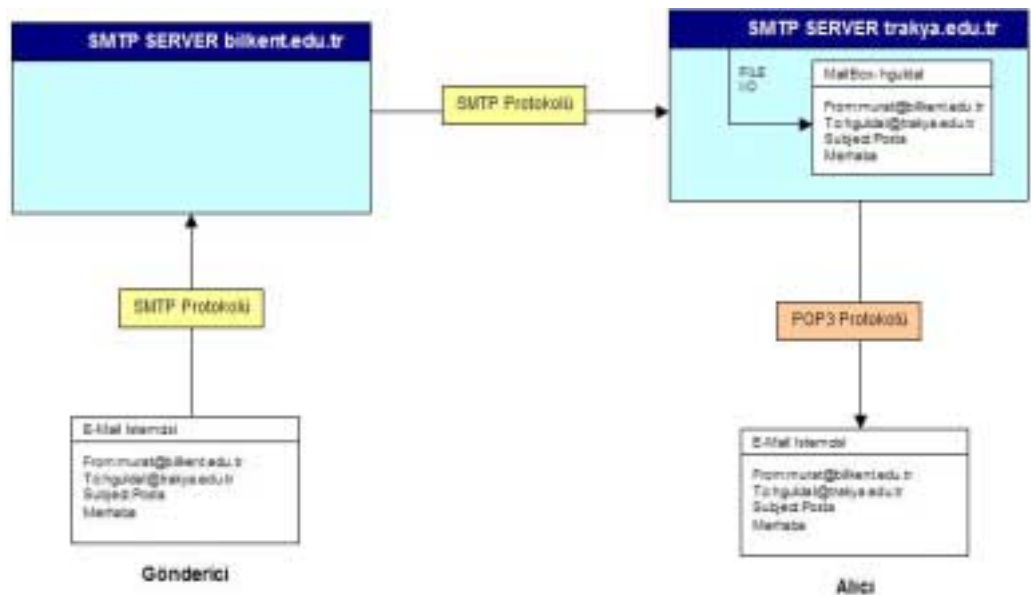
Durumlu protokol kullanan sunucular, işlemlerle ilgili bağlantının durumu, çalışmakta olan işler, bu işlerin durumu v.s. gibi bilgileri tutar. Genel olarak bu durum bilgisi için bellek kullanılır ve sistem kaynakları harcanır. İstemci durumlu protokol kullanan sunucu ile bağlantısını kestiğinde bu durum bilgisi silinmeli ve oturum koparılmalıdır.

Durumu olmayan protokoller küçük hacimlidir. Bu protokolleri kullanan sunucular tamamlanmış işlemler ve işler hakkında hiçbir bilgi tutmazlar. Bir istemci, durumu olmayan protokol kullanan sunucu ile olan bağlantısını kestiğinde hiçbir verinin silinmesine ya da oturumun koparılmasına gerek kalmaz. Durum bilgisinin kaydının tutulması sayesinde sunucu üzerine daha az yük biner ve sunucular işlemleri daha çabuk işleyebilirler.

2.4 Elektronik Posta Eriřim Paradigmaları

Elektronik posta erişiminde temel olarak üç model vardır. Bunlar Online Model, Offline Model ve Disconnected Model’lerdir.

Offline model günümüzde en yaygın olarak kullanılan elektronik posta erişim yaklaşımıdır. POP3 protokolü, bu modele göre tasarlanmıştır. Bu modelde bir istemci, periyodik olarak sunucuya bağlanır, posta kutusundaki tüm elektronik postaları kendi bilgisayarına çeker ve daha sonra bunları sunucudan siler. Bundan sonra, elektronik postalar üzerindeki tüm işlemler, istemcinin kendi bilgisayarında olup biter. Bu modelde, sunucunun görevi sadece elektronik postaları saklamak ve istediğinde istemciye göndermektir. Şekil 2.1 Offline Model’in çalışma şeklini göstermektedir. Bu modelde sunucuların üzerine çok fazla yük binmez. Bu yüzden günümüzde en yaygın olarak kullanılan modeldir. Bu avantajı ile birlikte bu modelin en büyük dezavantajı, elektronik postaların istemcinin bilgisayarına çekildikten sonra sunucudan silinmesi ve sadece istemci makinede kalmasıdır. Bu, istemci makineye olan bağımlılık anlamına gelir. Kullanıcının başka bir bilgisayardan elektronik postalarına erişme şansı bulunmaz. Bazı durumlarda istemci, postaları çektikten sonra sunucuda kopyalarını bırakabilir bu tür çalışmaya “Pseud-Online” denir.



Şekil 2.1. Offline Model’de Elektronik Posta Gönderme-Alma Süreci

Online model ise yaygın olarak uzak dosya sistemi protokollerini kullanır. Bu modelde temel olarak, istemci uygulama bir sunucu makinedeki posta kutusunu yönetir. Sunucuya bağlantı, oturum yoluyla yönetilir. Posta kutusundaki elektronik postalar sadece sunucuda saklanır ve istemciye gönderilmez. İstemci, gerek duyduğunda istediği bilgileri sunucudan kendi bilgisayarına çeker. Bu modelde doğal olarak tüm yük sunucu üzerine biner. Herşey sunucu üzerinde olup biter. Bu yüzden çok fazla sistem kaynağı kullanılır. Bu, büyük bir dezavantajdır. Buna rağmen en büyük avantajı elektronik postaların sunucuda saklanmasıyla dolaylı kullanıcının, istediği herhangi bir yerden ve herhangi bir bilgisayar sisteminden sunucuya bağlanıp posta kutusuna ulaşabilmesidir.

Disconnected model bu iki modelin karışımıdır. Kullanıcı elektronik postalarını kendi bilgisayarına çeker, bunlar üzerinde sunucu ile bağlantı olmaksızın işlemler-değişiklikler yapar. Daha sonra başka bir zamanda bu elektronik postaları sunucuya postalar. Bu, farklı istemciler kullanıldığında senkronizasyon problemini demektir ve bu yüzden her bir mesaj için benzersiz tanımlayıcılar kullanılır. Tablo 2.1, Modellerin çeşitli yönlerden karşılaştırmasını göstermektedir.

Tablo 2.1 Elektronik Posta Erişim Modellerinin karşılaştırılması

Özellik	Offline Model	Online Model	Disconnected Model
Birden Fazla İstemci Kullanabilme Desteği	Hayır	Evet	Evet
Minimum Sunucu Bağlantı Zamanı	Evet	Hayır	Evet
Minimum Sunucu Sistem Kaynağı Kullanımı	Evet	Hayır	Hayır
Minimum istemci disk kaynağı kullanımı	Hayır	Evet	Hayır
Birden fazla uzak posta kutusu kullanım Desteği	Hayır	Evet	Hayır
Bağlantı olmadan Elektronik Postalar üzerinde işlem	Evet	Hayır	Evet

2.5 Elektronik Posta Dökümanlarının Yapısı

Bir elektronik posta metni en temel anlamda bir bilgisayar ağı üzerinde bir noktadan diğerine hareket eden elektronik bir belgedir. Sonuçta, ortada bir iletişim olduğundan bu metnin önceden tanımlanmış göndericinin ve alıcının tanıdığı ortak bir formatta gönderilmesi gerekir. Günlük hayatımızda tanıdığımız birine bir posta gönderirken de böyle değil midir ? Biz mesajımızı bir kağıda yazar ve onu bir zarfın içine koyarız. Zarfın üzerinde gönderici adresini belli bir yere, alıcı adresini belli bir yere yazarız. Bu bilgilerin nereye ve nasıl yazılacağı, bizler için artık ortak standart haline gelmiştir. Bu düşünceden hareketle, elektronik posta metinleri ve daha da genel anlamda Internet üzerinde dolaşan elektronik dökümanlar (HTML belgeleri,vs..) için ortak standartlar belirleme gereği duyulmuştur. 1982 yılında D.Crocker, bu ortak standartların belirlendiği ilk doküman olan RFC 822’i yayınlamıştır. Bu RFC üzerinde ilerleyen yıllarda çeşitli revizyonlar yapıldı. Günümüzde Elektronik Posta metinlerinin formatının nasıl olması gerektiği Nisan 2001 yılında P. Resnick tarafından yayınlanan RFC ile kesin bir şekilde tanımlanmıştır.

Bir elektronik posta metni iki temel parçadan meydana gelir

1.Başlık Bilgileri

2.Gövde

Başlık bilgilerinde, elektronik posta hakkında bilgiler yer alır; örneğin, kimden gönderildiği, gönderilme tarihi, konusu, vs.

Buradaki bilgiler alanın adını takip eden “:” karakteri ve daha sonra alanın içeriği formatındadır. Her satır CRLF karakterleri ile sonlandırılır. Örnek olarak elektronik posta tarih ve konu bilgisi elektronik posta metni içinde aşağıdaki formatla karşı tarafa iletilir.

From: Merhaba CRLF

Date: Mon, 18 Apr 2005 10:41:26 +0300 CRLF

Alan bilgilerinin nasıl kodlanması gerektiği ve uyulacak kurallar RFC 2822’de ayrıntılı bir şekilde anlatılmaktadır.

Elektronik posta dökümanın gövdesi ise mesajın içeriğinin olduğu bölümdür. Bu bölümde karşı tarafa iletilmek istenen mesaj ASCII karakter kümesi kullanılarak yazılır. Yine her zamanki gibi her satır CRLF karakterleriyle sonlandırılır. Şekil 2.2’de elektronik posta metninin iç yapısı gösterilmiştir.

```
Return-Path: <owner-trakya@trakya.edu.tr>
X-Original-To: trakya-list-trakya
Delivered-To: hguldal@trakya.edu.tr
Received: by mail.trakya.edu.tr (Postfix)
        id AA262258FE; Mon, 18 Apr 2005 10:41:29 +0300 (EEST)
Delivered-To: trakya-list-trakya@trakya.edu.tr
Received: by mail.trakya.edu.tr (Postfix, from userid 123)
        id EE37F2589F; Mon, 18 Apr 2005 10:41:27 +0300 (EEST)
X-Original-To: trakya@trakya.edu.tr
Delivered-To: trakya@trakya.edu.tr
Received: from mail.trakya.edu.tr (localhost [127.0.0.1])
        by mail.trakya.edu.tr (Postfix) with ESMTP id 74E3D258F7;
        Mon, 18 Apr 2005 10:41:26 +0300 (EEST)
From: "=?ISO-8859-9?Q?Bilgi_DD=FElem_Daire_?=?ISO-8859-
9?Q?Ba=FEkanl=FD=F0=FD?=" <bidb@trakya.edu.tr>
Subject: =?ISO-8859-9?Q?DDinternet_Haftas=FD_E?=?ISO-8859-
9?Q?tkinli=F0ine_Davet?=
Date: Mon, 18 Apr 2005 10:41:26 +0300
Message-Id: <20050418073902.M56027@trakya.edu.tr>
X-Mailer: trakya.edu.tr webmail 20041109
X-OriginatingIP: 10.40.48.10 (bidb)
```

Şekil 2.2 Bir Elektronik Posta metninin iç yapısı

1992’ye gelinceye kadar elektronik postalar basit içerikli metin dizileri (ASCII dizileri) şeklinde olmuştur. N. Borenstein ve N. Freed, MIME adıyla 1341 numaralı bir RFC yayımladılar. Burada amaç, elektronik posta metinlerinin içerisine herhangi bir dosyanın eklenmesine olanak sağlamaktır. Bu düşünceyle birlikte çok parçalı (Multipart) elektronik posta dökümanları ortaya çıktı. Burada, bir elektronik posta metni, kendi içerisinde çeşitli “sınırlarla” (boundary) parçalara ayrılır ve eklenen dosyalar metin içerisindeki bu parçalara yerleştirilir. Şekil 2.3 eklentili bir elektronik posta dökümanın, kavramsal yapısını, Şekil 2.4 ise kodlanmasını göstermektedir.

Elektronik Posta Dökümanı

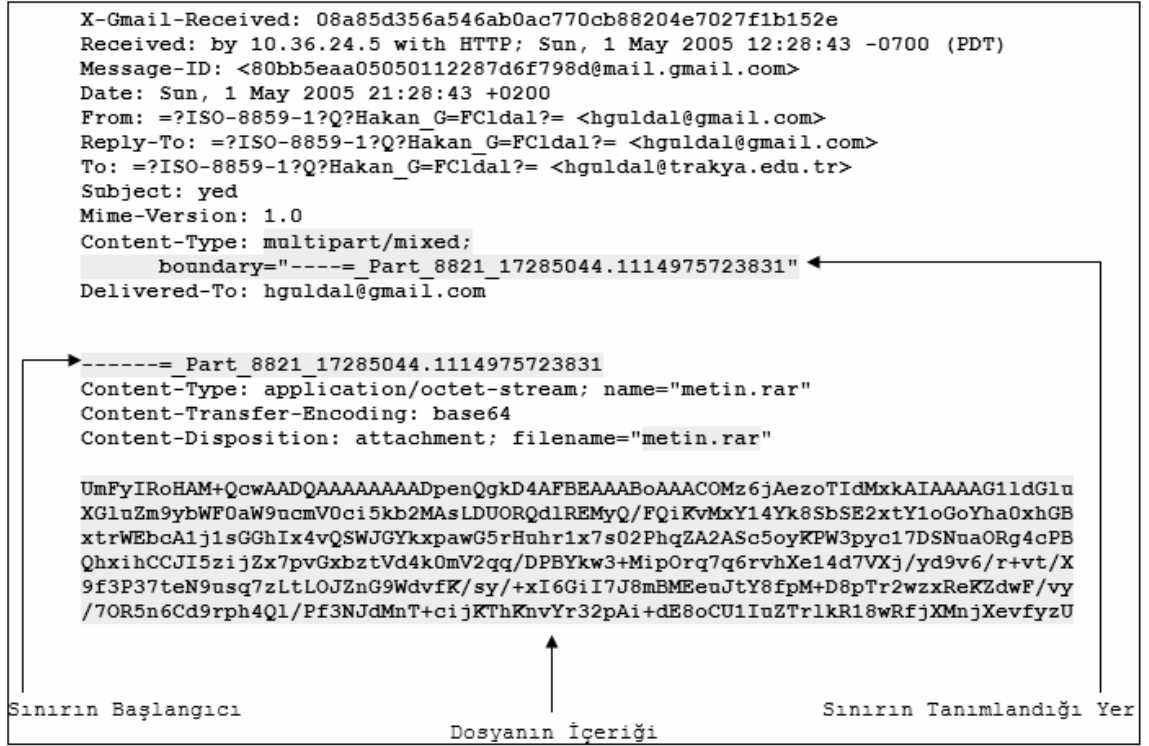
Başlık Bilgileri

Mesajın İçeriği

Parça 1 - resim.jpg

Parça 2 - muzik.wav

Sekil 2.3. Eklentili Elektronik Posta Metni



Şekil 2.4. Çok Parçalı Bir Elektronik Posta Metninin İç Formatı

3. POP3 PROTOKOLÜ

3.1 Tarihçe

POP (Post Office Protocol), en genel olarak, posta kutusuna gelen elektronik postaların, kullanıcı tarafından alınmasını sağlamak için tanımlanmış kurallar kümesidir.

Bir elektronik posta, SMTP (Simple Mail Transfer Protocol) protokolü kullanılarak, SMTP sunucusu tarafından kabul edilir ve alıcısının disk bölgesine koyulur. Bundan sonra, SMTP sunucusunun görevi biter. Elektronik posta, ayrılmış bu disk bölgesinde sahibi onu alıncaya kadar bekler. Elektronik postanın sahibi olan kullanıcı kendi bilgisayarındaki istemci yazılım (POP İstemcisi) aracılığı ile POP kullanarak sunucuya bağlanır ve postalarını kendi bilgisayarına çeker ve genellikle elektronik postayı sunucudan siler. Bundan sonra elektronik posta ile ilgili tüm işlemler kullanıcının kendi bilgisayarında olup biter.

POP ilk kez 1 Ekim 1984'de J.K Reynolds tarafından geliştirildi ve RFC 918 ile yayımlandı. Bu ilk sürümü 1.0 olarak adlandırılır. Daha sonra üzerinde bazı değişiklikler yapılarak 1 Şubat 1985'te M. Butler, J. Postel, D. Chase, J. Goldberger, J.K. Reynolds tarafından 2.0 sürümü geliştirilmiş ve RFC 937 ile yayınlanmıştır. 1 Kasım 1988 de, M.T. Rose Tarafından RFC 1081 ile Versiyon 3 geliştirildi. Mayıs 1996 a gelindiğinde J.Myers ve M.T. Rose Tarafından RFC 1939 ile Versiyon 3'te bazı düzeltmeler yapılarak protokol artık bir Internet standardı haline geldi (53 No'lu Internet Standartı). Protokol genellikle POP3 olarak anılır. POP3, günümüzde dünya üzerinde kullanılan en yaygın elektronik posta alım protokolüdür.

3.2 Genel Bakış

POP3 kullanımı kolay, açık ve güçlü bir protokoldür. Komut sayısı oldukça azdır. Protokolde tanımlanan komutlar genelde 3 veya 4 karakter uzunluğundadırlar. POP3 sunucuları genellikle varsayılan olarak 110 no'lu portu kullanırlar. Sunucu, 110 no'lu portu dinleyerek istemcilerin kendisine bağlanmasını bekler. Bağlanan bir istemciye sunucu tarafından ilk olarak, bir hoşgeldin mesajı gönderilir. Sunucu ile istemci arasındaki iletişim, bağlantı kopana kadar karşılıklı devam eder.

POP3'te komutlar büyük küçük harfe duyarlı değildirler ve istemcinin, gönderdiği her komutun sonuna CRLF karakterini eklemesi gerekir. Bazı komutlar parametre alırlar. Bu parametreler, komuta bir boşluk karakteri bırakılarak sunucuya gönderilirler ve uzunlukları 40 karaktere kadar ulaşabilir.

Sunucu, yanıtları her zaman sonuna CRLF ekleyerek istemciye gönderir ve bu yanıtların boyu 512 Bayt kadar olabilir. Gönderilen komutun sunucuda yarattığı etkiyi, istemciye belirtmek için protokolde iki durum göstergesi tanımlanmıştır. Bunlar, pozitif durum göstergesi "+OK" ve negatif durum göstergesi "-ERR" dir. Gönderilen komut sunucu tarafında hata üretmişse istemciye, başında negatif durum göstergesi "-ERR" bulunan bir yanıt döner. Eğer komutu başarıyla yürütülmüşse sunucudan başında pozitif durum göstergesi "+OK" bulunan bir yanıt döner. Durum göstergelerinin; "-ERR" ve "+OK", sunucu tarafından her zaman büyük harflerle gönderilmesi gerekir.

İstemci tarafından gönderilen komutlara sunucu tarafından gelen yanıtlar, bazen birden fazla satır tutabilir. Bu durum için protokolde bazı mekanizmalar sağlanmıştır. Sunucu, yanıtında, her satır sonuna CRLF kodunu ekler ve en son satırdan sonra sonlandırma karakterleri adı verilen bir nokta karakteri "." ve CRLF karakterlerini koyar. Bu sonlandırma karakterlerini algılayan istemci, sunucu tarafından gönderilen yanıtın bittiğini anlar. Birden fazla satırlı yanıtlardaki satırların herhangi birisi, sonlandırma karakterleri ile başlarsa bu durumda sonlandırma karakteri olarak CRLF.CRLF kullanılır. Bu yüzden, istemcilerin birden fazla satırlı olan yanıtlarda bu durumu göz önünde bulundurmaları ve kontrol etmeleri gerekir.

3.3 Durumlar

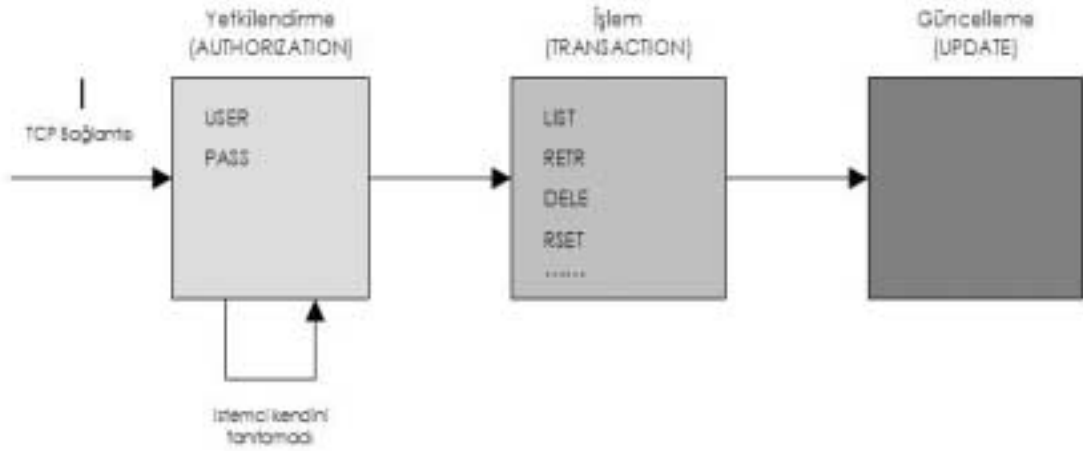
İstemci ile sunucu arasında gerçekleşen bir POP3 oturumu belli durumlardan meydana gelir. Bunlar, Yetkilendirme Durumu (AUTHORIZATION), İşlem Durumu (TRANSACTION) ve Güncelleme (UPDATE) durumudur.

Yetkilendirme durumunda istemci sunucuya kendini tanıtır. Bu işlem, iki olası yolla yapılır. Birinci yöntem kullanıcı adı-parola ikilisinin gönderilmesi, ikinci tanıtma yolu ise bazı POP3 sunucuları tarafından desteklenen opsiyonel bir komut olan APOP kullanmaktır. Burada parola gönderilmesi yerine daha farklı bir metod¹ kullanılır.

İstemci, sunucuya kendini tanıttığı zaman yetkilendirme durumu sona erer ve İşlem Durumu'na geçilir. Eğer yetkilendirme başarısız olmuşsa veya başka bir deyişle istemci kendini sunucuya tanıtamamışsa bu durum içinde kalınır ve işlem durumuna geçilmez. İşlem Durumu, istemcinin sunucudan belli eylemleri yerine getirmesi için taleplerini gönderdiği durumdur. Bu eylemler, posta kutusu hakkında bilgilerin istenilmesi, herhangi bir elektronik postanın silinmesi, herhangi bir postanın istemciye gönderilmesi vs. olabilir. İşlem durumunda kullanılan komutlar yetkilendirme durumunda kullanılmaz. İşlem durumu, istemcinin QUIT komutuyla sona erer. Bundan sonra Güncelleme durumuna geçilir. Burada ise, sunucu, kullandığı kaynakları serbest bırakır ve bir hoşçakal mesajı göndererek TCP bağlantısını koparır.

İstemci tarafından belirli bir süre komut gönderilmezse sunucu bağlantıyı otomatik olarak sonlandırır. Bu süre, sunucu tarafından belli bir zamanlayıcı kullanılarak takip edilir ve sunucudan sunucuya göre değişiklik gösterebilir. Fakat en az 10 dakika olmak zorundadır. Süre dolduğunda sunucu Güncelleme durumuna geçmeden ve herhangi bir mesaj göndermeden direkt olarak TCP bağlantısını koparır. Şekil 3.1'de POP3 oturumunun durumları gösterilmiştir.

¹ APOP komutu anlatılırken daha ayrıntılı incelenecektir.



Şekil 3.1. İstemci ile Sunucu arasındaki POP3 Oturumundaki Durumlar

3.3.1 Yetkilendirme Durumu

İstemci, POP3 sunucusuna bağlandığı zaman ilk olarak sunucu tarafından bir “hoş geldin” mesajı ile karşılanır ve böylece sunucu ile istemci arasında bir POP3 oturumu başlamış olur. Bu oturumda ilk olarak Yetkilendirme (AUTHORIZATION) durumunun içine girilir. Bu aşamada istemci kendisini sunucuya tanıtmak zorundadır. Yetkilendirme durumu, tanıma işlemi başarılı olduysa veya istemci QUIT komutu gönderirse sona erer. Yetkilendirme durumunda gönderilen QUIT komutuyla sunucu başka bir duruma geçmeden “hoşçakal” mesajı göndererek TCP/IP bağlantısını koparır. İstemcinin kendini tanıma işlemi başarılı olursa sunucu tarafından, posta kutusu diğer erişimlere kapatılır. Sadece bu POP3 oturumu artık postalar üzerinde erişim hakkına sahiptir ve sunucu bunu pozitif durum göstergesi (+OK) ile başlayan bir açıklama yanıtı ile istemciye bildirir. Sunucu bundan sonra İşlem durumuna geçer.

3.3.2 İşlem Durumu

İstemci sunucuya kendini tanıttıktan sonra posta kutusundaki elektronik postalar üzerinde işlem yapmaya yetkili kılınır ve talepler sunucu tarafından kabul edilir. Sunucudaki posta kutusunda elektronik postalar, geliş sıralarına göre sırayla tutulurlar ve dolayısıyla bir sıra numarasına sahiptirler. İstemci, herhangi bir elektronik posta üzerinde işlem yapmak isterken bu sıra numarasını kullanır. Örnek olarak, istemcinin talepleri sunucuya, posta kutusundaki 3 numaralı elektronik postayı sil veya 2 numaralı elektronik postayı bana gönder veya 4 numaralı elektronik postanın bilgilerini gönder gibi olur. Belli bir elektronik posta üzerinde işlem yapan bu tip taleplerde, bu komutlara parametre olarak işlem yapılmak istenen elektronik postanın sıra numarası verilir.

İstemcinin bazı talepleri sadece belli bir elektronik posta ile ilgili değil tüm posta kutusu ile ilgili olabilir. Örnek olarak posta kutusundaki tüm postaların tek tek boyutları veya posta kutusundaki postaların kapladığı alan sunucudan istenebilir.

İstemci, İşlem durumunda da Yetkilendirme durumunda olduğu gibi QUIT komutu göndererek POP3 oturumunu sonlandırmak isteyebilir. Bu durumda sunucu Yetkilendirme durumundan farklı olarak, bu komutu alır almaz TCP bağlantısını koparmak yerine POP3 oturumunda ilk önce Güncelleme durumuna geçer.

3.3.3 Güncelleme Durumu

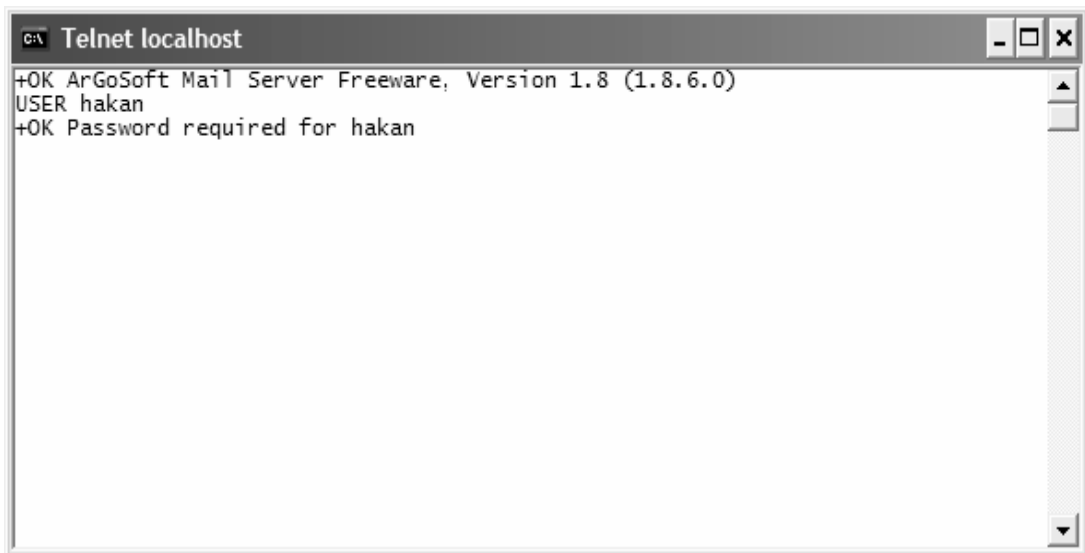
Sunucu ile İstemci arasındaki POP3 oturumundaki en son aşama Güncelleme durumudur. İstemci, İşlem durumundayken gönderdiği QUIT komutu ile bu duruma geçilir. POP3 sunucularında istemci, herhangi bir elektronik postayı silmek istediğinde sunucu bu elektronik postayı ilk önce “silinmiş” olarak işaretler; kalıcı olarak silmez. Bu bir geri dönüşürme mekanizmasıdır. Fakat sadece İşlem durumunda çalışır. Böylece istemci daha sonra, silinmiş olan bu elektronik postaları İşlem durumundayken geri alabilir. İstemci tarafından İşlem durumu sırasında gönderilen QUIT komutuyla Güncelleme durumuna geçildiğinde sunucu, silinmiş olarak işaretlenen tüm elektronik postaları kalıcı olarak siler. POP3 oturumunun ilk başında başka erişimlere kapadığı posta kutusunu başka erişimlere

tekrar serbest bırakır. İstemciye bir “hoşçakal” mesajı gönderir ve POP3 oturumunda kullandığı sistem kaynaklarını çalıştığı sisteme iade ederek istemci ile arasındaki TCP/IP bağlantısını sona erdirir.

3.4 Komutlar

3.4.1 USER Komutu

İstemci, postalar üzerinde işlem yapabilmek için ilk önce kendisini sunucuya tanıtmak zorundadır. Bu tanıma yöntemlerden biri *USER* ve *PASS* komut kombinasyonlarını kullanmaktır. İstemci, sunucuya kullanıcı adını *USER* komutuna bir boşluk bırakıp yazarak gönderir. Sunucu tarafından olumlu yanıt, pozitif durum göstergesi, olumsuz yanıt ise negatif durum göstergesi kullanılarak istemciye gönderilir. Olumsuz yanıtın sebebi gönderilen kullanıcı adıyla posta sunucusunda kayıtlı olanın eşleşmemesidir. Bazı posta sunucuları gönderilen kullanıcı adı ile kendinde kayıt olanlar arasında eşleşme yapmazlar bu durumda gönderilen her *USER* komutuna olumlu yanıt verirler. Şekil 3.2’de istemcinin gönderdiği *USER* komutu ve sunucu tarafından gelen yanıt gösterilmiştir.

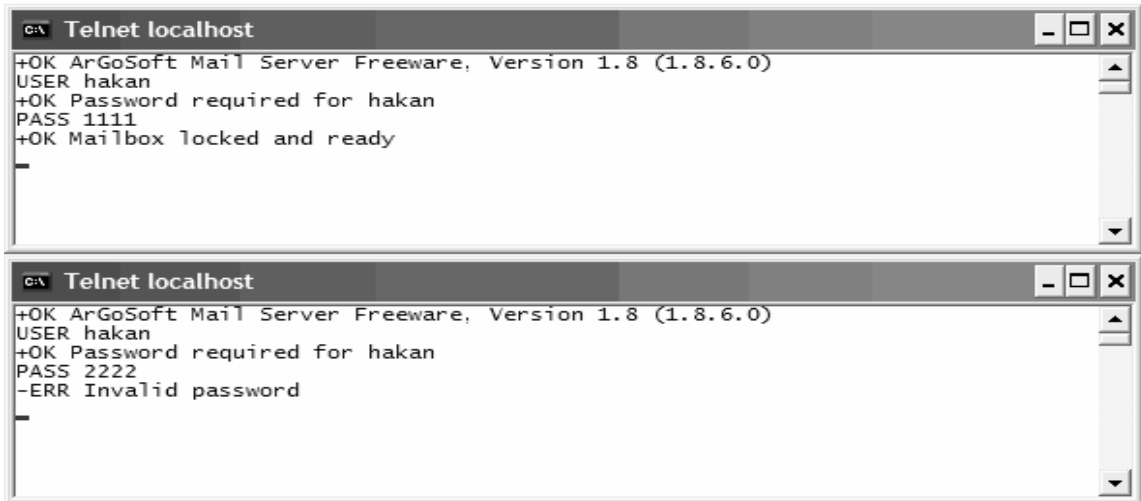


Şekil 3.2. İstemcinin *USER* komutuna sunucunun verdiği yanıt

3.4.2 PASS Komutu

İstemci tarafından *USER* komutu ile kullanıcı adı gönderildikten sonra ikinci aşama bu kullanıcı adına karşılık parolanın da gönderilmesidir. İstemci, *PASS* komutuna bir boşluk bırakır, parolayı yazar ve sunucuya gönderir. Sunucu, *PASS* komutunu algıladıktan sonra parametresindeki parolayı, bir adım önce aldığı kullanıcı adı ile birlikte kendinde kayıtlı olan kullanıcı adı-parola ikilisiyle karşılaştırır. Eğer gönderilen kullanıcı adı ve parola çifti kendindeki kayıtlı olan kullanıcı adı parola çiftiyle aynı ise yetkilendirme işlemi başarıyla tamamlanmıştır. Bundan sonra sunucu, istemciye yanıtın olumlu olduğunu bildiren pozitif durum göstergesi (+OK) ile başlayan bir mesaj gönderir. Şekil 3.3'de sunucunun gönderdiği pozitif durum göstergeli yanıt gösterilmiştir. Böylece istemci ve sunucu arasında oluşturulan POP3 oturumunda Yetkilendirme durumu sonlanıp İşlem durumuna geçilmiş olur.

İstemci tarafından gönderilen kullanıcı adı-parola çifti sunucuda kayıtlı olanlarla uyuşmazsa sunucu, istemciye negatif durum göstergesi (-ERR) başlayan olumsuz bir yanıt gönderir bu durumda istemci, kendisini sunucuya tanıtamamış olur ve POP3 oturumu yine Yetkilendirme durumunda kalır. Şekil 3.3'te sunucunun istemciye gönderdiği olumsuz yanıt gösterilmiştir.



Şekil 3.3. PASS Komutuna sunucunun gönderdiği olumlu ve olumsuz yanıtlar

3.4.3 APOP Komutu

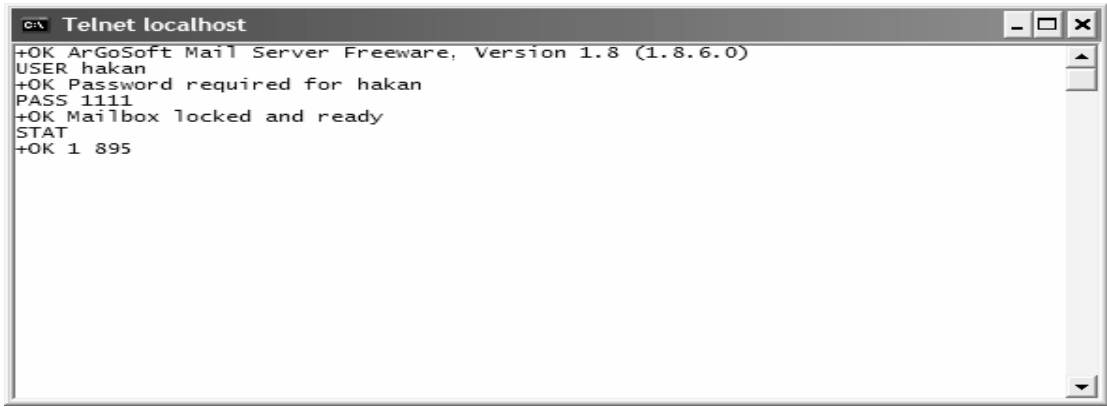
POP3 oturumundaki Yetkilendirme durumu sırasında istemci, sunucuya kendini tanıtmak için kullanıcı adı, parola kombinasyonlarını kullanırken bilgisayar ağı üzerinde parolanın, istemciden sunucuya doğru serbestçe yol almasından dolayı güvenlik açısından bazı sıkıntılar doğabilir. Bu amaçla kullanıcı adı-parola yöntemine alternatif POP3'e ikinci bir yetkilendirme mekanizması koyulmuştur.

Bu yöntemde, öncelikle sadece istemci ile sunucunun bildiği gizli bir parola belirlenir. İstemci, sunucuya bağlandığında sunucu, istemciye "hoşgeldin" mesajına ek olarak bir karakter dizisi gönderir. Bu karakter dizisi, her bağlantıda farklı olmak zorundadır (UNIX işletim sisteminde çalışan sunucunun proses numarası, sistem saati, sunucu adı'nın <prosesno-saat@sunucuadı> formatında göndermesi buna örnektir). Bu karakter dizisini alan istemci daha önceden belirlenmiş ortak parolayı da bu diziye ekler ve MD5 (RFC1321) algoritmasını uygulayarak şifreler ve daha sonra APOP komutuyla parametre olarak kullanıcı adından sonra parola eklenerek şifrelenmiş bu karakter dizisini sunucuya geri gönderir.

Sunucu, aldığı bu bilgiyi deşifre eder, gönderdiği karakter dizi ve ortak parola ikilisi ile karşılaştırır. Sonuç aynı ise yetkilendirme başarıyla tamamlanmış olur ve İşlem Durumu'na geçilir. Bu mekanizma kullanıcı adı-parola yetkilendirme yöntemine göre nispeten güvenlidir. En azından gizli veriler şifrelenerek iletilir. Fakat aynı zamanda risklidir. Ortak parolanın gizliliği çok önemlidir. İşlev olarak da diğerine göre daha kullanışsızdır. Bu yüzden POP3 sunucularının bir kısmı bu mekanizmayı desteklemezler. İnternet yaygınlaştıkça üzerinde iletilen verilerin güvenliğinin sağlanmasına yönelik çalışmalar artmıştır ve çeşitli yöntemler geliştirilmiştir. İletim güvenliği, POP3 istemci veya sunucularından çok bu tür iletim güvenliği sağlayan uygulamaların sorumluluğuna bırakılmıştır.

3.4.4 STAT Komutu

STAT komutu, sunucudan posta kutusu hakkındaki istatistiki bilgileri istemek için kullanılır. İşlem Durumunda kullanılan komutlarından birisidir. İstemci sunucuya bu komutu, herhangi bir parametre olmaksızın gönderir. Sunucudan gelen yanıt daima olumludur. Şekil 3.4'te istemcinin gönderdiği *STAT* komutu ve sunucudan gelen yanıt gösterilmiştir. Gelen yanıtın başında pozitif durum göstergesi *+OK* bulunur bir boşluktan sonra posta kutusundaki toplam posta adedi ve yine bir boşluktan sonra elektronik postaların sunucu üzerindeki posta kutusunda toplam kapladığı alan bayt olarak belirtilir.



Şekil 3.4. *STAT* Komutu ve Sunucu tarafından gönderilen yanıt

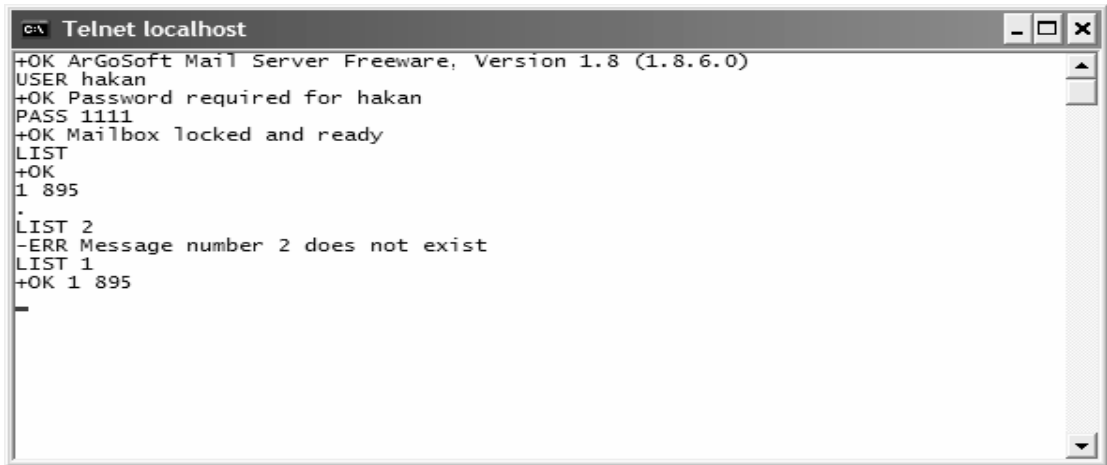
3.4.5 LIST Komutu

Sunucudan, posta kutusundaki elektronik postaların listesini istemek için kullanılan komuttur. İsteğe bağlı olarak parametre ile de kullanılabilir. Parametresiz kullanımında sunucu tarafından tüm elektronik postaların listesi istemciye gönderilir. Parametre ile kullanımında ise bilgi alınmak istenen elektronik postanın sıra numarası komuta eklenerek gönderilir. Sunucu, istemciye sadece bu parametrede belirtilen sıra numarasına sahip elektronik postanın bilgilerini gönderir.

Parametresiz kullanımda sunucudan gelen yanıt birden fazla satır içerir. Satırlar CRLF karakteriyle birbirinden ayrılmıştır. İlk satırın başında, pozitif durum göstergesi ile toplam mesaj adedi belirtilir ve her bir elektronik postanın sıra numarası ve boyutu ayrı bir satır olarak istemciye gönderilir.

Parametrelili kullanımda dönen yanıt tek bir satırdan oluşur. Bu yanıt ta pozitif durum göstergesi, sonrasında bir boşluk, elektronik postanın sıra numarası ve yine bir boşluk bırakılarak elektronik postanın kapladığı alan bayt cinsinden belirtilir.

Parametrelili kullanımda, gönderilen elektronik postanın sıra numarasına, sunucu tarafından posta kutusu içinde bulunmazsa sunucu istemciye hata oluştuğunu belirten ve negatif durum göstergesi ile başlayan bir yanıt gönderir. Şekil 3.5'te LIST komutunun parametrelili ve parametresiz kullanımı ile sunucudan gelen yanıtlar gösterilmiştir.



```

c:\ Telnet localhost
+OK ArGoSoft Mail Server Freeware, Version 1.8 (1.8.6.0)
USER hakan
+OK Password required for hakan
PASS 1111
+OK Mailbox locked and ready
LIST
+OK
1 895
.
LIST 2
-ERR Message number 2 does not exist
LIST 1
+OK 1 895

```

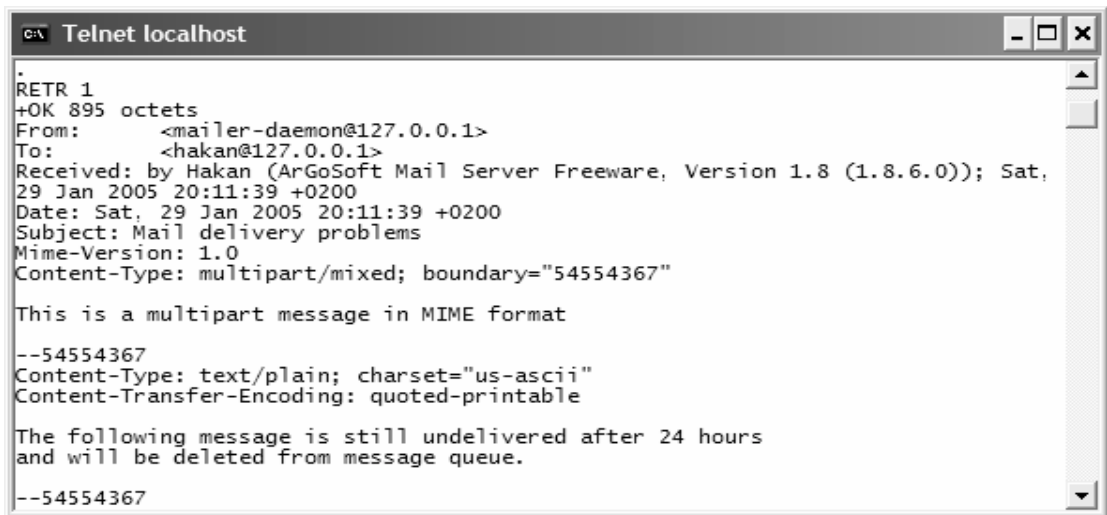
Şekil 3.5. LIST Komutu ve Sunucu tarafından gönderilen yanıt

SMTP sunucusu tarafından kabul edilen elektronik postalar kullanıcıya ayrılan disk bölgesine kopyalanırlar. Elektronik posta üzerinde SMTP sunucusunun işi, artık bu aşamada sona ermiştir. Elektronik postalar, bu disk alanında sahibi onu alıncaya kadar beklerler. İstemcinin, gelen bu elektronik postaları alma işlemi, sunucuya gönderilen *RETR* komutu sayesinde olur. Elektronik postalar, sunucuda, posta kutusu denilen kullanıcıya ait disk alanında geliş sıralarına göre bir numara verilerek saklanırlar.

İstemci, bilgisayarına çekmek istediği postanın numarasını *RETR* komutuna parametre olarak ekleyerek sunucuya gönderir. Bu komutu alan sunucu o sıra numarasına sahip postayı istemcinin bilgisayarına transfer etmeye başlar. Transfer etme işleminde, veriler birden fazla satırlar halinde gönderilir. Her olumlu yanıtın başına konulduğu gibi, burada da ilk satırın en başına pozitif durum göstergesi + OK konulur ve yine her satırın sonunda CRLF özel karakterleri bulunur. Postanın bitiminde en son satırın sonuna sunucu sonlandırma karakteri (.CRLF) koyar bu karakteri algılayan istemci postanın tamamıyla kendi makinesine transfer olduğunu anlar.

İstemci posta kutusunda olmayan bir posta numarasını parametre olarak gönderdiğinde bu, sunucu tarafında hata meydana getirir. Sunucu bu durumu negatif durum göstergesi -ERR ile başlayan ve hata mesajı ile devam eden bir yanıt ile istemciye bildirir.

Elektronik postaların boyutu bazen megabaytlar tutabilir ve bu yüzden transfer süresi uzun olur. İstemciye transfer edilen elektronik postanın formatı, Internet Mesaj Format Standardına (RFC 2822) göredir ve ASCII karakter dizilerinden oluşmuştur. Şekil 3.6'da RETR komutuna sunucunun gönderdiği yanıt gösterilmiştir. İstemcinin bilgisayarına ulaştıktan sonra posta, istemcinin bilgisayarında disk üzerinde herhangi bir metin tabanlı dosya haline gelir bütün denetim istemciye aittir. İçeriğinin yorumlanması ekrana gösterilmesi gerektiğinde silinmesi tamamıyla istemci tarafında meydana gelir.



```

C:\ Telnet localhost
.
RETR 1
+OK 895 octets
From: <mailer-daemon@127.0.0.1>
To: <hakan@127.0.0.1>
Received: by Hakan (ArGoSoft Mail Server Freeware, Version 1.8 (1.8.6.0)); Sat,
29 Jan 2005 20:11:39 +0200
Date: Sat, 29 Jan 2005 20:11:39 +0200
Subject: Mail delivery problems
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary="54554367"

This is a multipart message in MIME format

--54554367
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: quoted-printable

The following message is still undelivered after 24 hours
and will be deleted from message queue.

--54554367

```

Şekil 3.6. RETR Komutuna sunucudan yanıt ve elektronik postanın içeriği

3.4.7 DELE Komutu

Elektronik postalar istemci tarafından sunucudan alındıktan sonra sunucuda kalan kopyalarının silinmesi gerekir. Fakat bazı durumlarda postalar, istemci tarafından alındıktan sonra sunucu tarafında kopyaları kalabilir. Bu seçenek POP3 protokolünde sağlanmıştır. Silme işlemi posta gönderildikten hemen sonra sunucu tarafından otomatik olarak yapılabilirdi fakat daha esnek olması açısından bu işlem istemciye bırakılmıştır. İstemci eğer, elektronik postaların sunucuda kopyalarını bırakma seçeneğini kullanırsa bu tür çalışma şekline “Pseud-Online” denir.

DELE komutu sunucuda bulunan kullanıcının disk bölgesinde veya başka bir deyişle posta kutusundaki herhangi bir postayı silmesine olanak verir. Çalışma mantığı *RETR* ile hemen hemen aynıdır. İstemci, silinmek istenen postanın sıra numarasını sunucuya parametre olarak gönderir. Sunucu bu komutu aldığı anda silinmek istenen postayı ilk baştan kalıcı olarak silmek yerine “Silinmiş” olarak işaretler. Böyle olmasının sebebi istemcinin daha sonra bu işlemi geri alabilmesine olanak sağlamaktır. İstemci silinmiş olarak işaretlenmiş elektronik postaları oturumun ancak İşlem durumunda geri alabilir. POP3 oturumunun Güncelleme durumuna geçildiğinde sunucu silinmiş olarak işaretlenmiş tüm elektronik postaları kalıcı olarak siler.

İstemci tarafından gönderilen *DELE* komutu sunucu tarafından başarıyla işlendiğinde, istemciye pozitif durum göstergesi *+OK* ile başlayan bir yanıt gönderilir. Eğer gönderilen komut sunucu tarafında hata üretmişse istemci *-ERR* ile başlayan bir cevapla bilgilendirilir:

İstemci : *DELE 2*

Sunucu : *+OK Message number 2 marked*

İstemci : *DELE 8*

Sunucu : *-ERR Message number 8 does not exist*

3.4.8 RSET Komutu

RSET komutu, İstemci tarafından *DELE* komutu gönderilerek silinmiş olarak işaretlenen elektronik postalardan bu işareti kaldırır. Bu yönüyle bir nevi geri almak mekanizmasıdır. Komut, herhangi bir parametre verilmeksizin gönderilir ve sunucu tarafından tüm silinmiş olarak işaretli elektronik postalara uygulanır.

3.4.9 QUIT Komutu

QUIT komutu adından da anlaşıldığı gibi istemcinin POP3 sunucusuyla kurduğu bağlantısını sona erdirmesini sağlayan komuttur. Sunucu, istemcinin gönderdiği bu komutu aldığı anda Güncelleme Durumuna geçer ve eğer istemci tarafından silinmiş olarak işaretlenen elektronik posta varsa bunları kalıcı olarak disk bölgesinden siler ve sonra istemci ile kurulmuş TCP/IP bağlantısı koparılır.

Sunucu, komutu işletirken bazı olası hatalarla karşılaşabilir. Örneğin istemci tarafından silinecek olarak işaretlenmiş bir posta, sunucu tarafından disk bölgesinden tamamen silinemeyebilir. Bu durumda istemci, hata mesajı gönderilerek bilgilendirilir.

3.4.10 Diğer Komutlar

Yukarıdaki standart komutların dışında POP3 protokolünde bazı opsiyonel komutlar da bulunur. Bunların kullanılıp kullanılmaması sunucu yazılımını geliştiricilerine bırakılmıştır.

Örnek olarak;

Opsiyonel bir komut olan *TOP* komutu, *RETR* komutuna işlev olarak çok benzerdir. Fakat farklı olarak, talep edilen elektronik postanın tümünü değil sadece başlık bilgilerini istemciye gönderir.

4. SMTP PROTOKOLÜ

4.1 Genel Bakış

SMTP ilk kez Kasım 1981’de J.Postel tarafından geliştirildi ve ilk kez RFC 788 ile tanımlandı. 1 Ağustos 1982 de yine aynı kişi tarafından RFC 821 ile revizyona ugradı. Son olarak Nisan 2001 de J. Klensin protokolde bazı değişiklikler yaparak bunu RFC 2821 ile yayınlamıştır. SMTP günümüzde artık dünya çapında bir Internet standardı haline gelmiştir.

Protokol açık ve anlaşılması oldukça kolaydır. Komut seti az ve işlevseldir. SMTP sunucuları bilgisayar sisteminde varsayılan port olarak 25 numaralı portu kullanırlar. Sunucu, istemcinin her talebine karşı göndereceği veriye 3 haneli bir yanıt kodu ile başlar. Bu özel yanıt kodları protokolde açık bir şekilde tanımlanmıştır. Bu yanıt kodları, talep edilen isteğin, sunucu tarafında nasıl bir etki oluşturduğu hakkında istemciyi bilgilendirme niteliğindedir.

Aşağıda örnek olarak sunucu tarafından gönderilen bazı yanıt kodları verilmiştir:

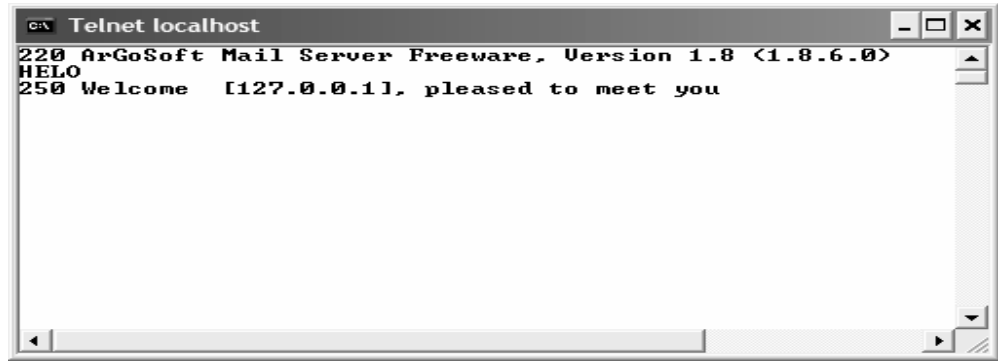
- 250** Talep Edilen Mail Eylemi Başarıyla Tamamlandı
- 251** kullanıcı Tanımlı Değil
- 500** Yazım Hatası, Komut Tanınamadı
- 501** Parametre veya Argümanlarda yazım hatası

SMTP’de de “CRLF.CRLF” karakter dizisine özel bir anlam yüklenmiştir. Bu 3 ardışık karakter data akışının sona erdiğinin göstergesi olarak kullanılır. Yazılan her elektronik posta metninin sonunda yollanırken veya alınırken bu 3 ardışık karakter bulunur.

4.2 Temel Komutlar

4.2.1 HELO Komutu

Anlamını ingilizcedeki “Hello” (Merhaba) kelimesinden almıştır. İstemci sunucuya bağlandıktan sonra “Merhaba” mesajı anlamına gelen “*HELO*” komutunu gönderir. Bu komutu alan sunucu sonra istemciye “hoşgeldin” niteliğinde bir yanıt gönderir. Şekil 4.1, HELO komutu ve sunucu tarafından bu komuta verilen yanıtı göstermektedir.

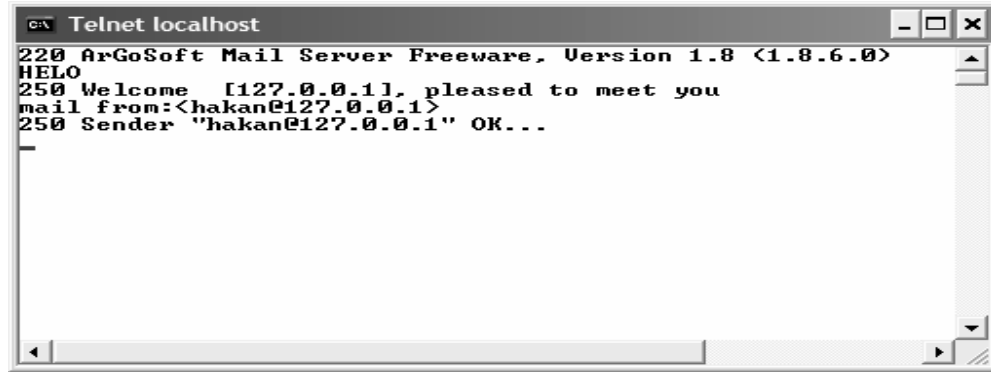


Şekil 4.1. HELO komutu ve sunucu tarafından gönderilen yanıt

4.2.2 MAIL Komutu

Elektronik postalarda en az 1 gönderici ve 1 alıcı adresi bulunur. *MAIL* komutu gönderici adresini sunucuya belirtmek için kullanılan komuttur. Elektronik posta adresi komuttan sonra bir boşluk bırakılarak parametre olarak yollanır.

Kullanımı, MAIL FROM:<user@host> şeklindedir.



Şekil 4.2 MAIL komutu ve sunucu tarafından gönderilen yanıt.

4.2.3 RCPT Komutu

Protokoldeki *RCPT* komutu adını İngilizcedeki “Recipient” (Alıcı) kelimesinin kısaltılmasıyla almıştır. Gönderilecek elektronik postanın alıcısını sunucuya belirtmek için kullanılır. Her elektronik postanın en az bir alıcısının bulunması gerekir. Birden fazla alıcı içeren elektronik postalarda her alıcı için komut tekrar gönderilebilir. Şekil 4.3’te RCPT komutunun kullanımı ve sunucu tarafından gelen yanıt gösterilmiştir.

Kullanımı, *RCPT TO: <kişi@hostname>* şeklindedir.

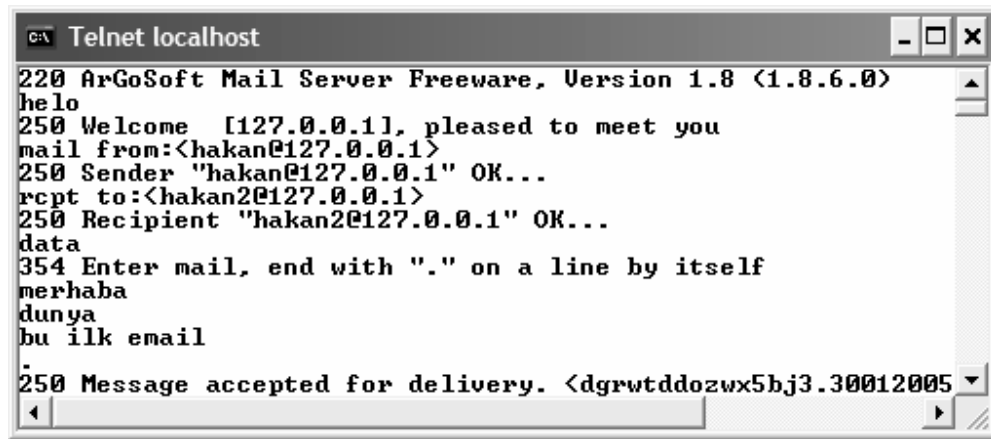
SMTP sunucusu gönderilen alıcı isminin kayıtlı olup olmadığına bakar ve bunun sonucuna göre istemciye olumlu veya olumsuz bir yanıt gönderir.



Şekil 4.3 RCPT komutu ve gönderilen olumlu yanıt

4.2.4 DATA Komutu

Elektronik postanın içeriği DATA komutu kullanılarak sunucuya iletilir. Komut istemci tarafından yollandıktan sonra sunucu, istemciye elektronik mesajı göndermesini ve bittiğinde sonuna CRLF.CRLF karakterlerini koymasını söyleyen bir yanıt gönderir. Bu aşamadan sonra istemci, mesajın içeriğini sunucuya gönderir. Mesajın en sonuna CRLF.CRLF karakterlerini koyar. Bu karakterleri algılayan sunucu mesajın sonuna geldiğini anlar ve elektronik mesaja, özel bir numara (*Message ID*) vererek daha önce *RCPT* komutuyla belirtilen alıcı adresindeki kullanıcının disk bölgesine koyar. Bu işlemlerin sonucunda istemciyi bir yanıt kodu göndererek bilgilendirir. Şekil 4.4'te DATA komutunun kullanımı ve sunucu tarafından gönderilen yanıtlar gösterilmiştir.



```

C:\ Telnet localhost
220 ArGoSoft Mail Server Freeware, Version 1.8 (1.8.6.0)
helo
250 Welcome [127.0.0.1], pleased to meet you
mail from:<hakan@127.0.0.1>
250 Sender "hakan@127.0.0.1" OK...
rcpt to:<hakan2@127.0.0.1>
250 Recipient "hakan2@127.0.0.1" OK...
data
354 Enter mail, end with "." on a line by itself
merhaba
dunya
bu ilk email
.
250 Message accepted for delivery. <dgrwtddozwx5bj3.30012005>

```

Şekil 4.4 *DATA* komutu ve sunucudan gelen yanıtlar

4.2.5 RSET Komutu

RSET komutu adını İngilizceki Reset kelimesinden almıştır. İşlevi, geçerli SMTP oturumunda yapılan işlemleri sıfırlamaktır.

4.2.6 QUIT Komutu

QUIT komutu, SMTP sunucusundan bağlantıyı koparmak için kullanılır. Bu komut sunucuya ulaştığında, sunucu istemciden bağlantısını koparır ve istemciye yanıt olarak bir “hoşçakal” mesajı gönderir. Bu aşamadan sonra SMTP oturumu sona erer. İstemci yeni bir SMTP oturumu açmak için tekrar sunucuya bağlanmak durumundadır.

5. KLASİK BİLGİ ERİŞİM SİSTEMİ MODELLERİ

5.1 Temel Kavramlar

Klasik Bilgi Erişim Modelleri'nde ana fikir, her bir metnin dizin terimleri veya içerik belirteçleri olarak adlandırılan anahtar kelimelerle ifade edilebileceğidir. Bir dizin terimi, temel olarak, bir metnin ana fikri hakkında bize ipuçları veren bir kelimedir. Böylece, dizin terimleri metinleri fihristlemek ve içeriklerini özetlemek için kullanılırlar. Genelde dizin terimleri, isimlerden meydana gelirler. Bunun nedeni, isimlerin kendi başlarına bağımsız anlamlara sahip olmalarıdır. Bundan dolayı akılda tutulmaları ve anlamsal olarak tanımlanmaları diğer türlere göre daha kolaydır.

k_i bir dizin terimi ve d_j bir metin olmak üzere $w_{i,j} \geq 0$ ağırlığı (k_i, d_j) ikilisinin birleşimidir ve bu ağırlık değeri metnin anlam içeriğini tanımlamada dizin teriminin öneminin ölçüsüdür.

Tanım

t sistemdeki tüm dizin terimlerinin sayısı ve k_i herhangi bir dizin terimi olmak üzere

- $K = \{k_1, k_2, \dots, k_t\}$ tüm dizin terimlerinin kümesidir.
- Bir ağırlık $w_{i,j} > 0$; herhangi bir d_j metni ve herhangi bir k_i dizin teriminin birleşimidir.
- Bir metin içerisinde geçmeyen bir dizin terimi için $w_{i,j} = 0$

Dizin terimi ağırlıklarının bazı durumlarda birbirlerinden bağımsız oldukları kabul edilir. Bunun anlamı, $w_{i,j}$ ağırlığının bilinmesi $w_{i+1,j}$ ağırlığının değeri hakkında bize bir şey söylemez. Fakat metnin içindeki dizin terimleri birbirleriyle tamamen ilişkisiz değildir. Dizin terimlerinin ağırlıkların birbirinden bağımsız olması işi basitleştirirken aynı zamanda doküman sıralamaların hızlı yapılmasına olanak sağladığı için de işi kolaylaştırır.

5.2 Mantıksal Model

Mantıksal model, Küme teorisi ve Boole cebir gibi matematiksel kavramlara dayanan bir bilgi erişim modelidir. Küme kavramının oldukça sezgisel olmasından dolayı Mantıksal Model, Bilgi Erişim modelleri içinde kullanıcılara, anlaşılması kolay bir çerçeveye sunar. Ayrıca bu modelde sorgulamalar, kesin anlamlara sahip mantıksal ifadelerle yapılır. Kendi özünden gelen basitlik ve güçlü biçimsellik nedeniyle Mantıksal Model bir Bilgi Erişim modeli olarak bu alanda yerini almıştır.

Fakat ne yazık ki bilgi erişim sürecinde mantıksal model çeşitli sınırlamalardan dolayı bazı durumlarda yetersiz kalır .

İlk olarak kendi bilgi erişim stratejisi herhangi bir ilişki ölçeği kullanmaz sadece ikili bir karar verme ölçütüne dayanır. Örnek olarak bir metin bir konuyla ya ilgilidir ya da değildir sadece iki durum söz konusudur. Böylece aslında Mantıksal Model bir Bilgi Erişim modelinden daha çok Veri Erişim (Data Retrieval) modeli olarak da değerlendirilebilir.

Modeldeki bir diğer sıkıntı, mantıksal ifadelerin kendi kesin anlamlarına sahip olmasından dolayı herhangi bir bilgiyi bir mantıksal ifadeye çevirme gerekliliğini ortaya çıkarmasıdır. Bu yüzden pek çok kullanıcı bu işlemi gerçekleştirmede zorluklar yaşar ve sorgu isteklerini mantıksal ifadeler türünden yazmada başarılı olamazlar.

Bu sınırlamalarına karşın Mantıksal Model ticari doküman veritabanı sistemlerinde hala baskın olarak kullanılan modeldir ve ayrıca bu alana yeni giriş yapacak olanlar için iyi bir başlama noktasıdır.

Mantıksal modele göre bir dizin terimi bir metnin içinde ya vardır veya yoktur. Buradan çıkacak sonuç, modelde, dizin teriminin ağırlığı ikili bir değer olarak kabul edilir. ($w_{i,j} \in \{0,1\}$) . Bir q sorgusu dizin ifadeleri ile birlikte üç bağlaç **değil**, **veya** ve **ve** nin birleştirilmesi ile oluşur.

Sonuç olarak Mantıksal modelin önemli avantajları modelin arkasında yatan temiz biçimsellik ve doğasında bulunan basitliktir. Bu avantajlarına rağmen en önemli

dezavantajı dizin terim ağırlığının esnek olmamasından dolayı bu modele dayalı uygulamalarda çok fazla doküman veya çok az sayıda dokümana erişilmesidir.

5.3 Vektör Model

Vektör modeli (Salton ve Lesk, 1968; Salton, 1971), mantıksal modelde, bir dokümanın bir dizin terimi ile ya ilgilidir ya da değildir düşüncesi veya başka bir deyişle dizin terimlerinin ağırlıklarının sadece 0 ve 1 gibi iki değer alabilmesi durumunun esnetilmesine olanak sağlayan bir çerçeve kurulmasını önerir. Modelde bu iş, sorgularda ve dokümanlarda dizin terimlerine ikili olmayan (non-binary) değerlerin atanması yoluyla başarılmaya çalışılır. Dizin terim ağırlıkları en sonunda, sistemde saklanan her bir doküman ile kullanıcı sorgusu arasındaki “*Benzerlik dereci*” lerini hesaplamada kullanılır. Erişilmiş dokümanların benzerlik derecelerine göre büyükten küçüğe doğru sıralanması ile vektör model, dokümanların hangi sorgu ifadesine kısmen uyduğunu gözönünde bulundurur. Temel sonuç olarak vektör modelde, derecelenmiş doküman, mantıksal modeldekinden daha doğru cevaplar verilmesine olanak sağlar.

Dizin terimlerinin ağırlıkları çok farklı şekillerde hesaplanabilir. Salton ve McGill’in çalışmalarında (Salton ve McGill, 1983) terim ağırlığı hesaplama teknikleri incelenmiştir. Burada bunlar detaylı bir şekilde incelenmeyecektir. Bunun yerine, en etkili terim ağırlığı hesaplama tekniklerinin arkasındaki ana fikir üzerindeki durulacaktır. Bu fikir aşağıdaki gibi toplama tekniklerine dayanan temel prensiplerle ilgilidir.

C nesneler kümesi ve A grubunun belirsiz bir tanımı verilmiş olsun, basit bir kümeleme algoritmasının amacı C kümesindeki nesneleri iki gruba ayırmak olabilir: birincisi A grubuyla ilgili olarak oluşturulan nesneler, ikincisi A grubuyla ilişkili olarak oluşturulmayan nesneler. Belirsiz tanım deyişi burada hangi nesnelerin A grubunda olduğuna hangilerinin olmadığına kesin bir karar vermek için yeterli bilginiz olmadığı anlamındadır. Örneğin birisi BMW fiyatıyla karşılaştırılabilecek arabaların kümesi A’yı arıyor olabilir. Burada karşılaştırılabilirin anlamı tam olarak açık olmadığından A grubu için açık bir tanım yoktur. Daha sofistike gruplama algoritmaları bir grubun nesnelerini, onların özelliklerine göre ayırmaya çalışırlar. Örneğin bir doktorun kanser tedavisi gören

hastaları 5 sınıfa ayrılabilir: ölümcül, ileri seviye, metaztas, teşhis edilen ve sağlıklı. Burada problem yeni gelen bir hastanın bu sınıflardan hangisine dahil edileceğidir. Yine de burada söz konusu olan sınıflama probleminin sadece en basit halidir. Çünkü tüm gereken sadece hangi dokümanların konuyla ilgili hangilerinin konuyla ilgisiz olduğuna karar vermektir.

Bilgi Erişim problemini sınıflandırmanın bir çeşidi olarak görmek için, Salton'un ilk çalışmalarına bakılacak olunursa. Dokümanlar C grubunun objeleri olarak ve kullanıcının sorgusu A grubunun belirsiz tanımlamaları olarak düşünülebilir. Bu senaryoda, Bilgi Erişim problemi, hangi dokümanların A grubunda olup olmadığını kararlaştırma problemine indirgenir. Bir gruplama (kümeleme) probleminde iki temel konu çözülmelidir. Birincisi, kişi A grubundaki nesneleri hangi özelliklerinin daha iyi tarif ettiğini belirlemelidir. İkincisi kişinin A grubundaki nesneleri, C grubunda kalan diğer nesnelerden hangi özelliklerinin daha iyi ayırt ettiğini belirmelidir. İlk grup özellikler küme içi benzerlikler için bir ölçüm sağlarken ikinci grup özellikler kümeler arası farklar için bir ölçüm sağlar. Sonuç olarak Vektör modellerinin temel avantajları şöyle sıralanabilir:

- 1) Terim ağırlığını ölçülmesi tarama performansını artırır.
- 2) Kısmi eşleştirme stratejisi yaklaşık sorgu koşullarında dokümanların taranmasına imkan sağlar
- 3) Dokümanlar sorguya benzerlik derecelerine göre sınıflandırılır.

Basitliğine rağmen vektör modeli genelde esnek bir bilgi erişim stratejisidir. Pek çok sayıda alternatif metotlar vektör modeliyle kıyaslanmıştır fakat genelde vektör modeli ya daha iyi yada en az bilinen alternatifleri kadar iyidir. Ayrıca basit ve hızlıdır. Bu nedenlerden dolayı bu günlerde vektör modeli popüler bir bilgi erişim modelidir.

5.4 Olasılık Yaklaşımlı Model

Daha sonraları ikili bağımsız tarama (BIR-Binary Information Retrieval) modeli olarak adlandırılan, olasılık yaklaşımlı model ilk kez Robertson ve Spark Jones (Rao vd., 1992) tarafından ortaya atılmıştır. Temel fikir şöyledir: Verilen bir kullanıcı sorgusu ile ilişkili dokümanları içeren bir dokümanlar grubu vardır. Bu doküman grubuna *ideal cevap kümesi* diyebiliriz. Bu ideal cevap kümesinin tanımı verilseydi dokümanlarını tararken

hiçbir problem olmazdı. Böylece sorgu işlemini, ideal cevap kümesinin özelliklerini belirtme süreci olarak düşünebiliriz. Problem, bu özelliklerin ne olduğunu tam olarak bilmememizdir. Tüm bildiğimiz anlamları, bu özellikleri karakterize etmek için kullanılan dizin terimlerinin olduğudur. Bu özellikler sorgu anında bilinmediğinden, ne olduklarını önceden tahmin etmek için çaba harcanmalıdır. Bu önceden yapılan tahmin, ilk doküman grubunu tararken kullanılan, olası tanımını oluşturmamıza olanak sağlar. İdeal cevap kümesinin olası tanımını geliştirmek için kullanıcıyla etkileşim o zaman başlar. Bu etkileşim şu şekilde işler:Kullanıcı taranan dokümanlara bakar ve hangisinin işe yarayıp yaramayacağına karar verir ve sisteme bunu belirtir. Bu sürece geribildirim süreci denir (Salton ve Buckley, 1990). Sistem daha sonra bu bilgiyi ideal cevap kümesinin tanımına yaklaşımda kullanır. Bu işlemi pek çok kez tekrarladıkça tanımlama, ideal cevap kümesinin gerçek tanımına yaklaşacak ve gelişecektir.

6. YAZILIMIN GERÇEKLEŞTİRİLMESİ

6.1 Genel Tanıtım

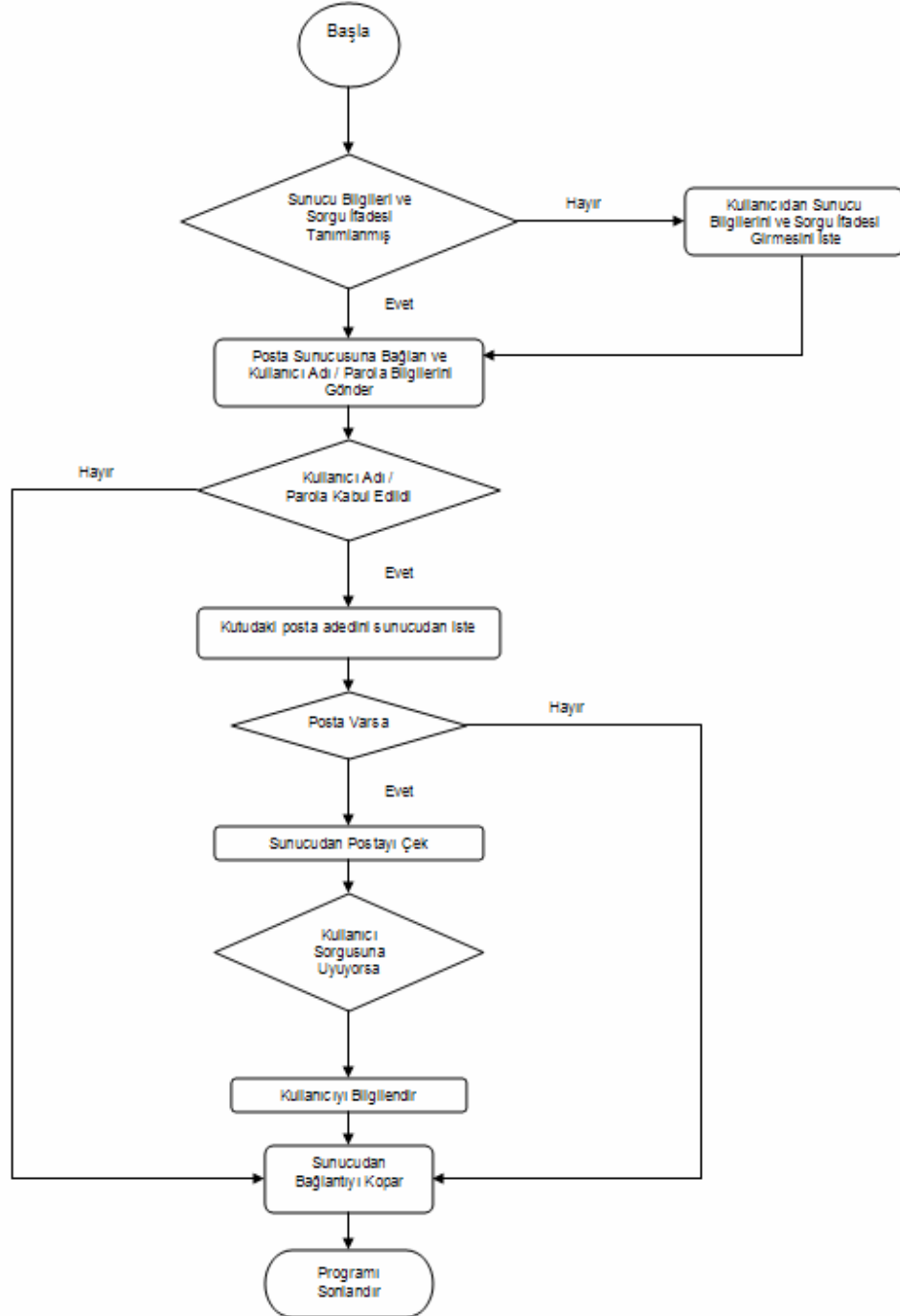
Bu çalışmada gerçekleştirmeye çalıştığımız yazılımın amacı, bir uzak posta sunucusuna POP3 protokolü kullanarak bağlanıp, kullanıcının hesabına gelen postalar üzerinde, yine kullanıcının belirlediği herhangi bir sorgu ifadesinin işletilmesidir. Buna göre kullanıcının gelen kutusuna düşen postalar yazılım tarafında, sorgu ifadesine uyanlar ve uymayanlar olarak iki sınıfa bölünür. İşlev olarak bakıldığında bu yazılımı bir filtre yazılımı olarak da kabul edebiliriz. Fakat, bilinen filtreleme uygulamaları genelde “tek kelime” sorgularına dayananırlar. Yazılımda ise “mantıksal sorgu” ifadeleri girmeye olanak verildiği için klasik filtre uygulamalarından daha esnek bir yapı sözkonusudur. POP3 sunucusundan çekilen elektronik posta, öncelikle okunur, başlık bilgileri (Kimden geldiği, tarihi, konusu) belirlenir . Kullanıcının önceden belirlediği sorgu ifadesi hemen uygulanır ve sonucunda elektronik posta, sorguya uyanlar ve uymayanlar olarak işaretlenir.

Yazılımı geliştirirken kullandığımız ortam işletim sistemi olarak Windows ve geliştirme aracı olarak da Visual Studio 6.0’dır. Posta sunucusuna bağlanıp kullanıcı hesabı üzerindeki postalarda işlem yaparken yazılım geliştirme ortamları için geliştirilmiş hazır POP3 istemci bileşeni kullanmak yerine Windows Socket temelinde, POP3 protokolü manuel olarak kodlanarak yazılımın içine yerleştirilmiştir. Bu açıdan bakıldığında yazılım aynı zamanda özel bir tür POP3 istemci uygulaması olarak da kabul edilebilir.

Yazılımda kullanılan sorgu ifadesi, VE, VEYA, DEĞİL mantıksal bağlaçlarının birleşiminden meydana gelir. Sorgu ifadesi kullanıcı tarafından önceden tanımlanır ve gelen postalar üzerine uygulanır.

Yazılımda elektronik posta eklenti dosyaları gözardı edilmiştir. Bunun nedeni, eklentiler, resim dosyalarından, ses dosyalarına, PDF dökümanlarından Word doküman formatı vb. gibi çok fazla çeşitlilik göstermesidir. Bu nedenle gelen posta, üzerinde sorgu işletilmeden önce eklentisinden ayrılır.

6.2 Yazılımın Genel Veri Akış Diyagramı



Şekil 6.1 Yazılımın Veri Akış Diyagramı

Yazılımın genel veri akışında, ilk önce sunucuya bağlanılır, ardından POP3 komutları kullanılarak gelen postalar okunur ve kullanıcı tarafından önceden belirlenen sorgu ifadesi uygulanarak, “sorguya uyanlar” ve “sorguya uymayanlar” olarak ayrılır ve en sonunda kullanıcıya bunlar gösterilir. Şekil 6.1’de bu işlemin akış diyagramı gösterilmiştir.

6.3 POP3 Sunucusuna Bağlanmak Ve Elektronik Postaların Alınması

Bilindiği gibi her işletim sistemi geliştirilirken üreticinin, kullanıcıların sistem üzerinde yazılım geliştirilebilmesine olanak sağlayan bir Uygulama Programlama Arabirimi (API) oluşturması gereklidir. Windows İşletim sistemi için de bu Uygulama Programlama Arabirimi geliştirilmiştir ve bu Windows API (WinAPI veya Win32API) olarak adlandırılır. Windows işletim sisteminin bu uygulama programlama arabirimi yüzlerce fonksiyondan oluşur ve bu fonksiyonlar genelde, yerine getirdikleri görevlere göre belli kütüphaneler şeklinde gruplanarak saklanırlar. Örnek olarak Shell32.DLL Windows işletim sisteminde kabuk işlemleri ile ilgili görevler yerine getiren fonksiyonları içerisinde barındırır. Buna benzer olarak Kernel.DLL çekirdek işlemleri, mmsystem.DLL ise multimedya işlemleri için gerekli fonksiyonları içerisinde bulundurulur.

İşletim sistemlerinde, aynı zamanda uygulama programlama arabiriminin bir parçası olan ve uzak bilgisayar sistemleri ile iletişim kurmasını sağlayan soket arayüzü mevcuttur. Windows’un soket arayüzü de Windows Socket API olarak adlandırılmıştır. Yazılımımızda Windows Socket API tabanlı, Winsock uygulama arayüzü kullanılmıştır. Winsock, herhangi bir uzak bilgisayarla TCP/IP veya UDP gibi ağ protokolleri kullanarak iletişim kuran uygulamalar geliştirmek için bir arayüzdür. Bu uygulamalar bir istemci uygulama olacağı gibi bir sunucu uygulama da olabilir.

Önceki bölümlerde anlatıldığı gibi POP3 bağlantılı ve durum bilgisi tutan bir Internet protokolüdür. Bir POP3 oturumu başladıktan sonra istemci tarafından gönderilen komutlara göre sunucu tarafında belirli durumlara geçilir. Aynı zamanda TCP/IP bağlantısı oturumunun sonuna kadar koparılmaz.

Yazılımımızda sunucuya bağlanmak ve POP3 oturumu oluşturmak için yukarıda da belirtildiği gibi Winsock arayüzü kullanılmıştır. Kullanıcı tarafından önceden tanımlanan

sunucu adresi ve port numarası ile TCP/IP protokolü kullanılarak karşı bilgisayara bağlanılır. Sunucu ile bağlantı sağlandığı zaman POP3 oturumu başlamış olur. Bağlantı sağlanamadığı halde kullanıcı uyarılır.

İstemci/Sunucu uygulamalar çift yönlü ve birbirini tamamlayan uygulamalardır. İstemci, belli bir kurala (protokol) göre sunucuya komutunu gönderir. Sunucu da aldığı komuta göre gerekli işlemleri yapar ve istemciye yanıt verisini gönderir. Söz konusu olan bilgisayarlar uzak bilgisayarlar olduğundan İstemci tarafından gönderilen her komut karşı sunucuya gitmesi belli bir süre ve aynı şekilde de sunucudan gönderilen yanıtın istemciye gelmesi de belli bir süre alır. Bundan dolayı bu tür çalışma şeklinde genellikle zaman aşımı mekanizması kullanılır. İstemci sunucuya komutunu gönderdikten sonra kendi tarafında bir zamanlayıcıyı başlatır ve saymaya başlar. Eğer süre önceden belirlenmiş olan zaman aşımı süresinden fazla tutarsa genelde sunucu ile bağıni koparır veya aynı komutu tekrar gönderir. Yazılımımızda da bu tür bir zaman aşımı mekanizması kullanılmıştır. Buna ilaveten gönderilen her POP3 komutundan sonra sunucuda gelen yanıtın kontrolü ve doğrulanması gerekir. Bu doğrulama olmaksızın diğer komut sunucuya gönderilmez. Böylece, ardışık komutların sunucuya gönderilmesi yanıt verilerinin alınması, bunların kontrolü ve aynı zamanda istemci tarafından gönderilen veriler üzerinde gerekli işlemlerin yapılması ile sunucu ile olan POP3 oturumu sona ermiş olur. Şekil 6.2 bu oturumun VB ile yazılmış kodunu göstermektedir.

```
Function POP3Oturum(Ok As Boolean, Hata As String)
On Error GoTo Hata
    Dim TmpPosta As String
    Dim s As String
    Dim R As String
    Dim IletiSayisi As Integer
    Dim i As Integer
    Dim TmpYeniPosta As String

    TmpPosta = YeniPostaSayisi

    If Not YanitVar(s) Then 'Eğer sunucudan yanıt gelmez ise
        Ok = False 'Hata var
        AnaForm.Socket.Close
        Durum = POP3_BaglantiYok 'durum: bağlantı yok
        Exit Sub
```



```

End If

Durum = POP3_User
AnaForm.Socket.SendData "USER " & KullaniciAdi & vbCrLf

If Not YanitVar(s) Then
    Ok = False
    AnaForm.Socket.Close
    Durum = POP3_BaglantiYok
Exit Sub
End If
Durum = POP3_Parola
AnaForm.Socket.SendData "PASS " & Parola & vbCrLf

If Not YanitVar(s) Then
    Ok = False
    AnaForm.Socket.Close
    Durum = POP3_BaglantiYok
Exit Sub
End If

If LCase(Left(s, 4)) = "-err" Then
    Ok = False
    AnaForm.Socket.Close
    Durum = POP3_BaglantiYok
Exit Sub
End If

Durum = POP3_MailAdedi

AnaForm.Socket.SendData "STAT " & vbCrLf

If Not YanitVar(s) Then
    Ok = False
    AnaForm.Socket.Close
    Durum = POP3_BaglantiYok
Exit Sub
End If

IletiSayisi = CInt(Mid$(s, 5, InStr(5, s, " ") - 5))

TmpYeniPosta = AlinmisPostaSayisi
Durum = POP3_MailAl

If IletiSayisi > 0 Then

    If IletiSayisi <> AlinmisPostaSayisi Then
        For i = 0 To IletiSayisi - 1

            AnaForm.Socket.SendData "RETR " & i + 1 & " 1" & vbCrLf

            If Not YanitVar(s) Then
                Ok = False
                AnaForm.Socket.Close
                Durum = POP3_BaglantiYok
                Exit Sub
            End If

        Next
    End If
End If

```

```

        AlinmisPostaSayisi = IletiSayisi

    End If

Else
    Ok = True 'hata yok
End If

AnaForm.Socket.SendData "QUIT" & vbCrLf

AnaForm.Socket.Close

Durum = POP3_BaglantiYok

Exit Sub

Hata:
AnaForm.Socket.Close
Durum = POP3_BaglantiYok
End Sub

```

Şekil 6.2 Sunucuyla POP3 protokolü ile iletişim kuran VB ile yazılmış fonksiyon

Yazılımın işlem bütünlüğünün sağlanması bakımından gönderilen her istemci komutuna karşın sunucu yanıtının kontrolü önemli bir iştir. Bu işi yerine getiren kod Şekil 6.3 ile gösterilmiştir.

```

Function YanitVar(m As String) As Boolean

    Dim basla As Single
    Dim Tmr As Single
    Dim a As String
    Dim n As Integer

    On Error GoTo Hata

    basla = Timer

    Select Case Durum
        Case POP3_MailAl

            While n <= 0
                DoEvents
                a = a & GelenVeri
                n = InStr(1, a, vbCrLf & "." & vbCrLf)

                'Eğer sunucudan veri gelmiyorsa zamanlayıcıyı başlat
                If Len(GelenVeri) = 0 Or GelenVeri = "" Then Tmr = Timer - basla

                If Tmr > ZamanAsimi Then

```

```

        'bu süre içinde sunucudan yanıt gelmezse
        GelenVeri = ""
        YanitVar = False
        Exit Function
    End If

    GelenVeri = ""

Wend

Case Else

    While Len(GelenVeri) = 0 Or GelenVeri = ""
        DoEvents
        a = GelenVeri

        Tmr = Timer - basla

        If Tmr > ZamanAsimi Then
            GelenVeri = ""
            YanitVar = False
            Exit Function
        End If

    Wend

End Select
m = a
GelenVeri = ""
YanitVar = True
Exit Function
Hata:
MsgBox Err.Description, vbCritical, LoadResString(Dil + 32)

End Function

```

Şekil 6.3 Sunucu yanıtını kontrol eden VB ile yazılmış fonksiyon

Şekil 6.3 ile gösterilen bu fonksiyon gönderilen her komuttan sonra çağrılır. Kodun içindeki döngülerde soket arayüzüne veri ulaşp ulaşmadığı her an kontrol edilir. Veri ulaştığı zaman döngüden ve fonksiyondan çıkılarak diğer komut istemciye gönderilir. Bu kısımda aynı zamanda zaman aşımı da kontrol edilir.

6.4 Elektronik Posta Metinleri Üzerinde Sorgunun İşletilmesi

Bir elektronik posta metni, basit olarak ifade etmek gerekirse sahip olduğu eklentisi gözardı edilirse özel formatı olan metin tabanlı bir dosyadır. Kullanıcının hesabındaki elektronik postalar sunucudan alındıktan sonra, yazılım tarafından okunur başlık bilgileri olarak adlandırılan kimden geldiği, konusu, gönderilme tarihi tespit edilir ve eğer varsa eklentisinden ayrılır bu aşamadan sonra elektronik posta metni üzerinde sorgu işletilmeye hazırdır.

Sorgu ifadesi, anahtar kelimeler ve bunların mantıksal işleçlerle (VE, VEYA, DEĞİL) birleştirilmesinden oluşan bir cümle olarak kabul edilebilir. Bu amaçla, sorgu ifadesinin tanımlanması için bazı kabullenmeler ve kurallar konulmuştur.

Tablo 6.1 Mantıksal İşleçler ve Yazılımdaki karşılıkları

Mantıksal İşleç	Yazılımdaki Karşılığı
VE	Ve
VEYA	Veya
DEĞİL	-

- İki anahtar kelimeyi bağlayan mantıksal bağlacın önünde ve arkasında birer boşluk karakteri olması gerekir. Örneğin; *elma veya armut* gibi.
- DEĞİL' in karşılığı olan “-” karakteri anahtar kelimelerin önüne gelmek zorundadır.örneğin ; *-elma* gibi.

Bir örnek verecek olursak elektronik posta metni içerisinde “trakya” ve “üniversite” kelimelerinin geçtiği fakat “fakülte” kelimesinin geçmeyenler aranmak istendiğinde girilecek olan ifade

trakya ve üniversite ve -fakülte

şeklinde olacaktır.

Bir diğer önemli nokta da; yazılımda, işlem önceliği soldan sağa olarak tercih edilmiştir. Yukarıdaki örneğe bakacak olursak ilk önce “*trakya ve üniversite*” işleme tabi tutulacak ve daha sonra buradan çıkan mantıksal sonuç (veya w ağırlığı) ile *-fakülte*’nin mantıksal sonucu ve bağlacı ile birleştirilerek tüm ifadenin mantıksal sonucu ortaya çıkacaktır.

```
Function SorguUygula(Str As String) As Boolean

    Dim kelime
    Dim s As Boolean
    Dim Parca() As Boolean
    Dim i As Integer

    kelime = Split(LCase(Trim(Kriter)), " ", , vbTextCompare)

    ReDim Preserve Parca(UBound(kelime))

    For i = 0 To UBound(kelime)
        If kelime(i) <> "ve" And kelime(i) <> "veya" Then

            If Left(kelime(i), 1) = "-" Then

                kelime(i) = Right(kelime(i), Len(kelime(i)) - 1)

                Parca(i) = Not (Str Like "*" & kelime(i) & "*")

            Else

                Parca(i) = (Str Like "*" & kelime(i) & "*")

            End If

        End If

    Next

    For i = 0 To UBound(kelime)

        Select Case kelime(i)

            Case "ve":

                Parca(i + 1) = Parca(i - 1) And Parca(i + 1)

            Case "veya":

                Parca(i + 1) = Parca(i - 1) Or Parca(i + 1)

        End Select

    Next

End Function
```

```
SorguUygula = Parca(UBound(kelime))  
End Function
```

Şekil 6.4 Yazılım içerisinde sorgu ifadesinin bir elektronik posta metnine nasıl uygulandığını gösteren Visual Basic kaynak kodu.

6.5 Sonuçların Kullanıcıya Gösterilmesi

POP3 sunucundan tüm elektronik postalar alındıktan ve sorgu ifadesi uygulandıktan sonra, sorgu sonucuna göre postalar yazılım tarafından işaretlenirler. Daha sonra tüm elektronik postaların bir liste halinde kim tarafından gönderildiği, konusu, tarihi gibi bilgileri gösterilir. Sorgu ifadesine uyan postalar kullanıcının dikkatini çekecek koyu bir yazı tipiyle ekranda listelenir.

7. TARTIŞMA VE SONUÇ

Sonuç olarak, bu çalışmamızda, kullanıcının bir POP3 sunucusunda saklanan elektronik posta dökümanları üzerinde, anahtar kelime tabanlı mantıksal sorgu ifadeleri işletilmesine olanak veren ve bunun sonucuna göre elektronik postaları sorguya uyan veya uymayan şeklinde sınıflandıran bir yazılım gerçekleştirilmesine çalışılmıştır.

Çalışmamızdaki sorgu ifadesinin işletilme mekanizması ve sonucundaki sınıflandırma, daha farklı platformlarda ve farklı döküman tipleri için de kullanılması düşünülebilir. Uygun yapısal değişikliklerle platform, masaüstü bilgisayar yerine bir sunucu bilgisayar, dökümanlar ise, posta sunucusunda saklanan ve POP3 protokolü yardımıyla erişilebilen elektronik postalar yerine bilgisayar ortamında “dosya” şeklinde saklanan elektronik posta metinleri olabileceği gibi ASCII tabanlı herhangi bir metin dosyası vb. de olabilir.

Internet üzerinde hergün elektronik posta şeklinde milyonlarca doküman dolaşmaktadır ve bu sayı yapılan hesaplamalara göre gün geçtikçe hızla artmaktadır. Bu dökümanların büyük bir kısmı alıcıları tarafından, gereksiz ve istenmeyen olarak nitelendirilen türdendir. Elektronik postaların, yazılımlar tarafından otomatik olarak sınıflandırılması probleminin çözümü günümüzde önemli bir çalışma alanıdır. Bu açıdan bakılacak olursa bu çalışma, bu problemin çözümüne katkıda küçük bir adımdır.

Diğer yandan anahtar kelimelere dayanan mantıksal sorgular günümüzde özellikle web için çok popülerdirler ve belki çoğu kullanıcının bilgi ihtiyacını karşılamaya yetecek kadar da güçlü görünürler. Fakat aynı zamanda salt kelimelere dayanmaları daha doğrusu kelimelerin anlamlarından soyutlanmış hallerine dayanmaları bir tartışma konusudur. Bu yüzden günümüzde, kullanıcıların aramalarını daha esnek hale getirecek ve arama sonunda daha sağlıklı ve kesin sonuçlar elde etmelerini sağlayacak olan doğal dil ve semantik tabanlı yeni yöntemler üzerinde çalışılmaktadır. Gelecekte, muhtemelen, arama yazılımları, kullanıcının ne elde etmek istediğini çok daha iyi tahmin eden ve buna göre çok daha kesin sonuçlar üreten, bir çeşit yapay zeka ile donatılmış sistemler olacaktır.

KAYNAKLAR

ACKERMAN R., 2003, *Vector Model Information Retrieval*,
<http://www.hray.com/5264/math.html>.

BAEZA-YATES R., RIBERIO-NETO B., 1999, *Modern Information Retrieval*, ACM Press.

BORENSTEIN, N., FREED N., 1992, *MIME (Multipurpose Internet Mail Extensions)*.

BRIN S., PAGE L., 1998, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*,
<http://www-db.stanford.edu/~backrub/google.html>.

CRISPIN, M., 2001, *Internet Message Access Protocol - Version 4*, RFC 2060 .

CROCKER D.H., 1982, *Standart For The Format of ARPA Internet Text Messages*, RFC 822.

CROFT W.B., 1995, *What Do People Want from Information Retrieval?*
<http://www.dlib.org/dlib/november95/11croft.html>.

GREY T., 1995, *Message Access Paradigms And Protocols*,
<http://imap.org/imap.vs.pop.html>.

KLENSIN, J., 2001, *Simple Mail Transfer Protocol*, RFC 2821.

LICKLIDER J.C.R., CLARK W., 1962, *On-Line Man Computer Communication*, MIT

MYERS, J., ROSE, M., 1994, *Post Office Protocol*, RFC 1725.

POSTEL J. B., 1982, *Simple Mail Transfer Protocol*, RFC 821.

RAO R., CARD S.K., JELLINEK H.D., MACKINLAY J.D., ROBERTSON G.G., 1992, *The Information Grid: A framework for building information retrieval and retrieval-centered applications. In Proc. Of the ACM Symposium on User Interface Software and Technology*, Monterey, CA, USA.

RESNICK, P., 2001, *Internet Message Format*, RFC 2822.

REYNOLDS, J. K., 1984, *Post Office Protocol*, RFC 918

ROBERTS L.G., 1966, *Towards a Cooperative Network of Time-Shared Computers*, MIT.

SALTON G., MCGILL M.J., 1983, *Introduction to Modern Information Retrieval*. McGraw Hill Book Co. New York.

SALTON G., LESK M.E., 1968. *Computer evaluation of indexing and text processing*. Journal of the Acm, 15(1):8-36.

SALTON G., 1971, *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

SALTON G., BUCKLEY C., 1990, *Improving retrieval performance by relevance feedback*. Journal Of American Society For Information Science, 41(4):288-297.

STANEK W.R., 1997, *Html Java Cgi VrmI Sgml – Unleashed*, Sams Net Co.

YERGEAU, F., 1998, *UTF-8, a transformation format of ISO 10646*, RFC 2279.
<http://www.tbd.org.tr/index.php?module=dict>.

ÖZGEÇMİŞ

Hakan GÜLDAL, 1977 yılında Edirne’de doğdu. İlk ve ortaöğrenimini Edirne’de tamamladıktan sonra Trakya Üniversitesi Fen Edebiyat Fakültesi Matematik Bölümü’nden 2001 yılında mezun oldu. Aynı yılın Aralık ayında Trakya Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü’ne Araştırma Görevlisi olarak atandı. 2002 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda yüksek lisansa başladı. Halen, Trakya Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü’nde, çalışmalarını sürdürmektedir.